CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Information Technology

Proceedings of the 9th Prague Embedded Systems Workshop

July 1-3, 2021
Horoměřice
Czech Republic

Editors:

doc. Ing. Hana Kubátová, CSc.
doc. Ing. Petr Fišer, Ph.D.
Ing. Jaroslav Borecký, Ph.D.

ISBN 978-80-01-06858-8

Message from the Program Chairs

The Prague Embedded Systems Workshop is a research meeting intended for the presentation and discussion of students' results and progress in all aspects of embedded systems design, testing, and applications. It is organized by members of the Department of Digital Design at Faculty of Information Technology (which is the youngest faculty) of the Czech Technical University in Prague (which is the oldest technical university in Central Europe). The workshop aims to enhance collaboration between different universities not only inside the EU. It will be based on oral presentations, mutual communication, and discussions.

There are three types of students´ submissions and presentations at PESW 2021:

- Full papers describing the student's original research. These papers were submitted to a standard reviewing process.
- Abstracts of authors' earlier published and successfully presented papers (at conferences, journals, etc.). These contributions were not reviewed; emphasis was put on the presentation and discussion.
- Student posters - abstracts of defended bachelor and master theses with subsequent presentation only.

Eight papers were accepted for PESW 2021 presentation, from which there were 3 full papers and 5 abstracts. Contributions from Czech Republic, France, Germany, Italy, and Poland were accepted this year.

The technical program is also highlighted by three keynote speakers in the areas of cryptography, machine learning, and design:

- Privacy Preserving Collaborative Learning.
  Speaker: Oana Stan, CEA, Systems Safety & Security laboratory, France
- Modular Arithmetic-based Circuits and Systems for Emerging Technologies and Applications: Deep Neural Networks and Cryptography.
  Speaker: Leonel Sousa, INESC Lisboa, Instituto Superior Técnico, Universidade de Lisboa, Portugal
- Component-Based Design by Solving Language Equations.
  Speaker: Tiziano Villa, Universita di Verona, Italy

Three technical sessions and two poster sessions were formed.

Last but not least we would like to thank to our sponsors (CTU in Prague, EATON, ASICentrum, SYSGO, CESNET, and STMicroelectronics).
Special thanks go to IEEE: IEEE Student Branch at Czech Technical University in Prague and IEEE Young Professionals, organizing student contest, and Czechoslovakia Section of IEEE.

Hana Kubátová and Petr Fišer

Committees

Workshop Chairs

Hana Kubátová, CTU in Prague (CZ)

Petr Fišer, CTU in Prague (CZ)

Programme Committee

A. Bernasconi, Università di Pisa (IT)

P. Bernardi, Politecnico di Torino (IT)

A. Bosio, École Centrale de Lyon (FR)

T. Čejka, CTU in Prague (CZ)

G. DiNatale, TIMA, Grenoble (FR)

P. Fišer, CTU in Prague (CZ)

J.L. Gaudiot, University of California, Irvine (USA)

K. Jelemenská, STU Bratislava (SK)

M. Jenihhin, Tallinn Univ. of Technology (EE)

L. Kekely, BUT, Brno (CZ)

P. Kitsos, TEI West. Greece (GR)

Z. Kotásek, BUT, Brno (CZ)

H. Kubátová, CTU in Prague (CZ)

F. Leporati, Univ. di Pavia (GR)

A. McEwan, University of Leicester (UK)

N. Mentens, KU Leuven (BE)

P. Mróz, University of Zielona Gora (PL)

V. Muthukumar, Univ. of Nevada, Las Vegas (USA)

M. Novotný, CTU in Prague (CZ)

A. Orailoglu, UC San Diego (USA)

Z. Plíva, TU Liberec (CZ)

J. Raik, Tallinn Univ. of Technology (EE)

O. Ryšavý, BUT, Brno (CZ)

J. Schmidt, CTU in Prague (CZ)

M. Skrbek, CTU in Prague (CZ)

B. Steinbach, TU Chemnitz (DE)

J. Strnadel, BUT, Brno (CZ)

R. Ubar, Tallinn Univ. of Technology (EE)

Z. Vašíček, BUT, Brno (CZ)

P. Velan, ICS MUNI (CZ)

M. Zachariášová, ASICentrum (CZ)

W. Zając, Jacob of Paradies University (PL)

Special Session on Network Security Chair

Tomáš Čejka, CTU in Prague (CZ)

Student Poster Session Co-Chairs

Tomáš Kolárik, CTU in Prague (CZ)

Jan Bělohoubek, CTU in Prague (CZ)

Organizing Committee

H. Kubátová, CTU in Prague (CZ)

P. Fišer, CTU in Prague (CZ)

J. Borecký, CTU in Prague (CZ)

M. Novotný, CTU in Prague (CZ)

R. Kinc, AMCA (CZ)

E. Uhrová, AMCA (CZ)

Contents

Keynotes

Semiconductor Innovation Supporting Industrial Trends

Speaker: **Roman Ludin**, *STMicroelectronics, Czech Rep.*

In this presentation you will learn about importance of semiconductor components innovation to the Industry 4.0 evolution. Quick time line lookback followed by outlook into incoming new semiconductor components brining high level of innovation and integration allowing designers to implement more smart and more autonomous systems.

Roman Ludin

Roman has played several key roles inside STMicroelectronics since joining the company in 2003. He is focusing on ST microcontrollers and microprocessors, supporting leading customers across in Europe. Currently Roman is driving team of Field Application Engineers which mission is to enable developers and designers of embedded applications to release their creativity and bring essential innovation into final products.

Privacy Preserving Collaborative Learning

Speaker: **Oana Stan**, *CEA, Systems Safety & Security laboratory, France*

Machine learning is feasible for various use cases; however, the traditional approach requires all training data locally. This is an issue when multiple institutions are willing to collaborate but cannot share sensitive data. Promising recent technologies for collaborative use of machine learning are Federated Learning or Private Aggregation of Teacher Ensembles (PATE). This talk will briefly present an ongoing research project funded by Defense research (PADR) of the European Union that aims to apply these concepts to real-world use cases and improved them with privacy-preserving enhancements such as homomorphic encryption.

Oana Stan

Dr. Oana Stan is a full-time researcher at CEA as a member of the Systems Safety & Security laboratory. Prior to her current position, she accomplished a Ph.D. in Computer Science on discrete optimization under uncertainty applied to the compilation of parallel programs for manycore architectures. Her main research activities from 2013 include the application of advanced cryptographic techniques for secure computation such as the homomorphic encryption, functional encryption or verifiable computing. She was involved in various R&D projects (national and European) related to the compilation for programs working on encrypted data as well as the deployment of secure computation methods in practical settings such as the analysis of energy data in smart cities and protection of data privacy for end-users in health-related applications.

Modular Arithmetic-based Circuits and Systems for Emerging Technologies and Applications: Deep Neural Networks and Cryptography

Speaker: **Leonel Sousa**, *INESC Lisboa, Instituto Superior Técnico, Universidade de Lisboa, Portugal*

Energy efficiency and limited power consumption are key aspects for the next-generation of integrated circuits and systems. Thus, together with the increase of performance, they should drive the design of new architectures and arithmetic units. Unconventional number systems, namely Residue Number Systems (RNSs), may hold the answer to these emerging challenges. RNS relies on the use of modular arithmetic to perform additions, subtractions and multiplications in parallel without any dependency between the RNS-digits, thus improving the energy efficiency. Due to a few limitations, such as conversion overheads and division, only recently have RNSs experienced a significant number of advances in its application to new domains, such as Deep Convolutional Neural Networks (DCNN) and cryptography. In this talk, we present a state-of-the-art overview concerning the use of the RNS not only to improve the performance of public-key cryptographic algorithms but also to make them more resistant to attacks. RNS for emerging post-quantum algorithms, namely the ones supporting lattice-based cryptosystems (LBCs), and Fully Homomorphic Encryption (FHE) are also covered in this seminar. The potential of RNS for the high-performance implementation of deep convolutional neural networks (DCNNs) is unveiled. A novel hardware implementation of RNS-based matrix multiplication useful for implementing DCNNs is discussed in this seminar.

Leonel Sousa

Leonel Sousa received his PhD in electrical and computer engineering from the Instituto Superior Técnico (IST), Universidade de Lisboa (UL), Lisbon, Portugal, in 1996. He is currently a Full Professor and Chair of the Electrical and Computer Engineering Department at the IST and a Senior Researcher with the Instituto de Engenharia de Sistemas e Computadores – Investigação e Desenvolvimento (INESC-ID), Lisbon, Portugal. He spent three months in Japan at the beginning of 2017 with a prestigious JSPS Invitation Fellowship for Research, and he has been a Visiting Professor at The Carnegie Mellon University (CM) in the fall semester of 2017/2018. He has given more than 30 keynote, invited talks and tutorials. He has authored or co-authored more than 250 papers, appearing in international journals and conferences, and edited five special issues of international journals. As professor, he has given several undergraduate and graduate courses, and supervised 15 PhD Theses.

His research interests include computer architectures, parallel computing, computer arithmetic, and multimedia systems. Prof. Sousa is a Senior Member of IEEE, Fellow of the IET, and a Distinguished Scientist of the ACM. He served as a member of the organization committee for several international conferences, and he is currently an Associate Editor and Editor-in-Chief of several renowned international journals, including two IEEE Transactions and the IEEE Access. He received several awards for the quality and impact of his scientific publications (DASIP, SAMOS, UL/Santander).

Component-Based Design by Solving Language Equations

Speaker: **Tiziano Villa**, *Universita di Verona, Italy*

An important step in the design of a complex system is its decomposition into a number of interacting components, of which some are given (known) and some need to be synthesized (unknown). Then a basic task in the design flow is to synthesize an unknown component that when combined with the known part of the system (the context) satisfies a given specification. This problem arises in several applications ranging from sequential synthesis to the design of discrete controllers. There are different formulations of the problem, depending on the formal models to specify the system and its components, the composition operators, and the conformance relations of the composed system vs. the specification. Various behavioral models have been studied in the literature, e.g., finite state machines and automata, omega-automata, Petri nets, process spaces process algebras; various forms of synchronous and asynchronous (interleaving/parallel) composition have been considered; the conformance relations include language containment and equality, and notions of simulation. In this talk we give an overview of the problem (a.k.a., the unknown component problem, or submodule construction, etc.), and we focus on its reduction to solving equations over languages, as a key technology for supporting synthesis of compositional systems. We survey the state-of-art and highlight open problems requiring further investigation.

Tiziano Villa

Tiziano Villa completed a Ph.D. in EECS in 1995 at the University of California, Berkeley. In 1997 he joined as a Research Scientist the PARADES Labs, Rome, Italy. In 2002 he became an Associate Professor at Universita di Udine, Italy. Since 2006 he is a Professor with the Department of Computer Science (DI), Universita di Verona, Italy. His research interests are in formal methods for electronic design automation, including logic synthesis, formal verification, automata theory and models of computation, discrete-event dynamic systems, supervisory control, cyber-physical and embedded systems. He co-authored three books: Synthesis of Finite State Machines: Functional Optimization (Kluwer/Springer - 1997 and reprint 2010), Synthesis of Finite State Machines: Logic Optimization (Kluwer/Springer - 1997 and reprint 2012), The Unknown Component Problem: Theory and Applications (Springer - 2012), and co-edited the book "Coordination Control of Distributed Systems" (Springer, 2015).

Secure Software Updates: Challenges and Solutions for Embedded IoT Systems

Florian Herbold, Andrea Reindl, Hans Meier,
Michael Niemetz, Stefan Krämer
Faculty of Electrical Engineering and Information Technology,
Ostbayerische Technische Hochschule (OTH) Regensburg
Regensburg, Germany

{florian.herbold, andrea.reindl}@st.oth-regensburg.de,
{hans.meier, michael.niemetz}@oth-regensburg.de,
stefan.kraemer@dzg-metering.de

Abstract. The invention of the internet made the development of intelligent networking of millions of embedded systems possible. This enabled smart buildings, power grids and cities as well as applications in the fields of health, agriculture and industry. These systems frequently perform safety-critical applications and operations. This makes it urgent to protect these sensible systems as effectively as possible. Especially firmware updates are often the weak point in the systems. If unauthorised persons gain access to the system during the update, malware can be injected or sensitive data can be read and stolen.
This paper describes the challenges of secure firmware updates. To protect an embedded system from potential attackers, the concepts integrity, authenticity and confidentiality have to be adhered during the update process. Otherwise, there is an increased risk of modifying or reverse engineering the firmware image. Likewise, inadequately protected software can enable the installation of third-party firmware as well as the installation of firmware on a third-party system. Threat prevention is presented with solutions derived from functional safety and IT security. Aspects of protection against errors in the transmission of updates and against attacks aiming to compromise the system are explained. Finally, a possible sequence of a secure update process is examined in detail for a real embedded system implementation. For this purpose, the preparation, transmission and installation of a firmware update in the bootloader are discussed.

Keywords. Firmware Update, Cryptography, Embedded Systems, Bootloader, Safety, Security, Software Protection, IoT

1 Introduction

Due to a lack of firmware updates, outdated software versions and security vulnerabilities remain. This allows attackers to exploit the same security flaws on different devices. Worst-case scenarios of devices with outdated firmware have already occurred. One example is the *Mirai botnet*, which infected many devices on different platforms, creating a large global botnet for Distributed Denial of Service (DDoS) attacks [1]. Thousands of routers belonging to the company *Deutsche Telekom* have already been targeted by this attack leaving customers temporarily without internet connection [2, 3]. To prevent such attacks it is necessary to develop secure firmware updates for a wide range of platforms and devices.

This paper provides an overview of methods to protect firmware updates. It focuses on software solutions to ensure the correct transmission of the update and to prevent accidental or intentional modification of the firmware. Solutions to protect against physical attacks, such as the use of Trusted Platform Modules (TPMs) [4], are not discussed.

The aim is to give the reader the basic knowledge about possibilities to protect firmware updates in order to develop software for secure updates. For this purpose, a specially developed firmware update process is presented, which enables secure firmware updates of a low power system for digital electricity meters wirelessly by using bluetooth and a smartphone.

Section 2 describes threats and concepts that can arise during secure firmware updates. Afterwards, section 3 outlines various solutions to protect firmware updates from attackers. A possible process of a secure firmware update in the bootloader of a commercially used system is presented in section 4. Finally, the results are discussed and conclusions are drawn in section 5.

2 Software Update Challenges

The large number of devices on the market do not allow firmware updates to be performed by the manufacturer or by an employee on site. It is therefore necessary to provide firmware updates that can be installed directly by the user. This in turn also involves risks which attackers can exploit to gain access to the system. Before further focusing on how firmware updates can be secured using the example of a commercial system, the most serious threats and significant key concepts during firmware updates are defined. [5]

2.1 Security Threats

Inadequately protected software may allow an attacker to steal the technology of the manufacturer and thus its intellectual property. Updates that are performed by the user make it necessary to secure the entire process way of the firmware update from the manufacturer to the final device. A weak security of the firmware update process allows an attacker to take advantage of several threats. The possible threats associated with firmware updates are described in Table 1. [5, 6]

Table 1: Security threats during firmware updates [5–8].

Threat ID	Name	Description
T-01	Alteration	(Partial) modification of the firmware distributed by the manufacturer.
T-02	Reverse engineering	Reconstructing the firmware in assembler or higher level languages to analyze the firmware.
T-03	Unauthorised firmware	Uploading firmware from an unknown source that may not have been developed for the device.
T-04	Unauthorised device	Installing the firmware of the manufacturer on an unauthorized device.

2.2 Security Concepts

Information and data are assets worth protecting [9]. Therefore, several concepts have been defined to protect the assets from intentional attacks against IT systems. The following concepts are particularly important for the security of firmware updates. These have to be applied to update systems securely and to reduce threats during an update. [7, 10]:

Confidentiality – Ensures that data of the firmware update can only be read or modified by authorised users. This concept applies both when accessing stored data and during data transmission. Applying symmetric or asymmetric encryption algorithms can fulfill this requirement.

Authenticity – Provides the properties of genuineness, verifiability and trustworthiness of data during a firmware update. This concept confirms a recipient that the data comes from a reliable source. Digital signatures, certificates or Message Authentication Codes (MACs) are used to create the necessary root of trust.

Integrity – During a firmware update data must not be manipulated unnoticed. For this purpose, all changes must be traceable. To detect these alterations of the firmware data during transmission, methods such as error detection and correction or message numbering are used.

Table 2 shows the threats that can be reduced by adhering to a concept. Consequently, disregarding any of the threats leads to an increased risk of an attack. For example this means that the risk of unauthorised firmware being installed on the system is increased if the authenticity is not maintained.
Of course, it would also be possible to checkmark the authenticity column for threat T-01 (Table 2). Authenticity can also ensure the integrity of the data. If the hash is incorrect or the signature is invalid, the firmware image could also have been altered. The integrity would have been compromised as a result. On the other hand, a breach of authenticity does not automatically mean a violation of integrity. It is possible that a correct transmitted firmware image is verified with the wrong hash function or with a wrong key for the digital signature. Consequently, the authenticity is violated but not the integrity of the data. For this reason, the use of one single concept is often not sufficient to obtain a functionally secure system. Different concepts must always be used to prevent attacks.

Table 2: Comparison of which security threats can be prevented by the concepts of IT security [7].

Threat ID	**Security Concepts**		
	Confidentiality	**Authenticity**	**Integrity**
T-01			✓
T-02	✓		
T-03		✓	
T-04		✓	

3 Solutions

Protecting software completely against attacks is not possible. Despite this, there are some effective methods to make system attacks more time-consuming and costly and thus less interesting for the attacker. The following sections analyze various security and safety methods in order to ensure secure firmware updates. [11]

3.1 Safety Solutions

The protection against errors during data transmission is part of the functional safety. The aim is to detect and correct errors in order to achieve a secure system state in every case during the firmware update. For this purpose, the most relevant methods are presented below [6, 11]:

Error Detection and Correction – Error detection is one of the most commonly used methods to ensure that the message was received correctly. Often used mechanisms for this matter are Cyclic Redundancy Checks (CRCs) or hashes, which are appended to the end of a message. The receiver recalculates the CRC value and compares the received value with the calculated one. The less frequent error correction can be helpful when retransmission is limited due to power consumption, transmission time or frequency. This can be archieved with the help of Hamming codes.

Message Numbering – Message numbering is used to prevent messages from being lost and to validate that they arrive in the correct sequence. During firmware updates especially the correct sequence of messages is important, because missing data makes the firmware unusable. Targetting this, a sequence number is appended to each message and then incremented by one. However, low transmission bandwidth or power consumption make retransmission difficult. Consequently, mechanisms for unordered transmission are required to process each properly received message.

Message Acknowledgement – Acknowledgement of each correct received message is required to detect the loss of a message. If the sender does not receive an acknowledgement after a specified period of time, an erroneous transmission is assumed and the message is resent. This ensures the correct sequence of messages. However, an incorrect implementation of a firmware update using message acknowledgement prevents further messages from being sent until an acknowledgement of the first packet is returned. The maximum transmission speed can not be used and the duration of the firmware update is extended.

Since protocols such as TCP/IP or CAN already implement the methods mentioned above, it is always an adequate solution to rely on existing protocols. They are well approved and widely used, which minimizes the risk of errors.

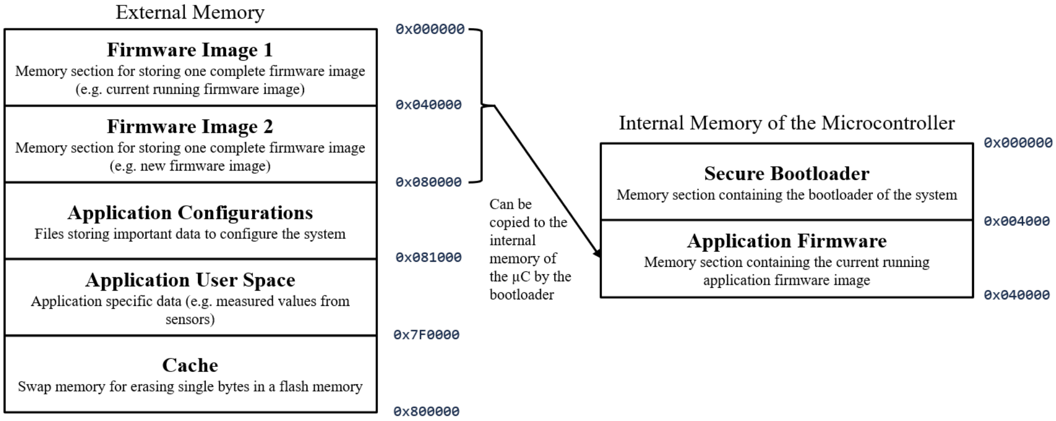

External Memory
Firmware Image 1
Memory section for storing one complete firmware image (e.g. current running firmware image)
Firmware Image 2
Memory section for storing one complete firmware image (e.g. new firmware image)
Application Configurations
Files storing important data to configure the system
Application User Space
Application specific data (e.g. measured values from sensors)
Cache
Swap memory for erasing single bytes in a flash memory
0x000000
0x040000
0x080000
0x081000
0x7F0000
0x800000
Can be copied to the internal memory of the µC by the bootloader
Internal Memory of the Microcontroller
Secure Bootloader
Memory section containing the bootloader of the system
Application Firmware
Memory section containing the current running application firmware image
0x000000
0x004000
0x040000

Figure 1: Example memory layouts for the internal and external memory of the system.

Furthermore, the protection of the system must be guaranteed to make sure the system can return to a safe state in the event of an error. This is why a backup copy of the current running image can be stored in an external memory. Therefore it is divided into several sections, which allows important configurations of the system, two different versions of firmware images and application-specific data to be stored. The new firmware image is stored in one memory section and the previous stable firmware image in the other one. If an error occurs during a firmware update and the program cannot start, the system is always capable to switch back to the previous firmware version and remains in a safe operating state. For this, the bootloader of the system is able to copy the stable firmware image located in the external memory into the internal application firmware section. Figure 1 illustrates the example memory layouts for dividing the external and internal memory into several sections. [9, 11, 12]

3.2 Security Solutions

To protect a firmware update against attacks that attempt to compromise the system, methods from the IT security are applied. Thereby, IT security is the property of a functionally secure system to only accept system states that do not lead to unauthorised information modification or acquisition [9]. To prevent these threats and especially to prevent accidental or intentional firmware modification, the following methods allow to perform a check of the transmitted data [7, 11, 12]:

Hashes – A hash creates a digital fingerprint to ensure that the firmware has not been modified. Hash functions make it unlikely to get the same hash value for different data. If the received and calculated hash are identical, the received and sent firmware is the same and data changes are excluded. However, since anyone can generate a hash, this method alone is not sufficient for ensuring the integrity of the firmware. If an attacker gains access to the firmware file and modifies it, it can calculate a valid hash and install the tampered firmware onto the system.

Digital Signatures – Digital signatures fix the problem of simply hashing the firmware by encrypting the calculated hash using an asymmetric method. The manufacturer signs the hash of the firmware with their private key which is only known to them. During firmware updates the bootloader of the system checks the firmware image by decoding the digital signature with the public key and afterwards checks the hash. The digital signature thus serves as a trust anchor for the firmware update. This prevents an attack described in the previous point, since the attacker does not know the private key.

Message Authentication Code (MAC) – A MAC is similar to a digital signature but uses a symmetric encryption method instead of an asymmetric one. Since a MAC is based on a symmetric method, the calculation time is significantly reduced compared to the method using digital signatures. As the encryption and decryption keys are identical, an attacker who appropriates the private key can also verify and compute the MAC. The safety of the system is thus compromised and the firmware can be altered analog to the solution using hashes.

The above outlined methods comply with the two security concepts authenticity and integrity in the system (Table 1). To additionally ensure confidentiality the firmware itself can be secured using a symmetric or asymmetric encryption method [7]. This means that the firmware cannot be altered (T-01) or reverse engineered (T-02, Table 1) by attackers [6].

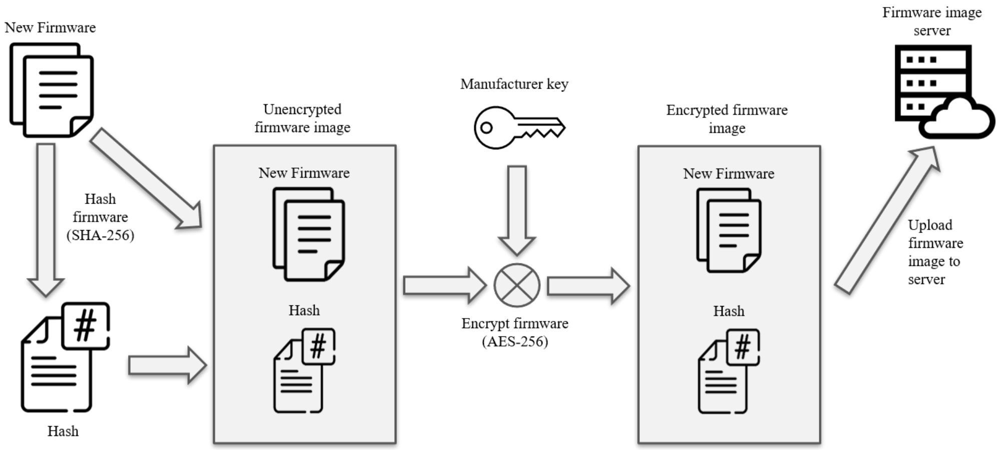


Figure 2: Generating a new firmware update on the side of the manufacturer.

4 Procedure of a Secure Update Process

The following section shows the usage of the described methods for a firmware update in a commercially used system. The module enables the transmission of meter readings of digital electricity meters through Bluetooth Low Energy (BLE) to a smartphone app [13, 14]. In addition to displaying the current power consumption or historical meter readings, the smartphone app also handles the transfer of a new firmware image in case of an available update.

4.1 Preparation and Transfer of a Firmware Update

Before the firmware update can be installed on the device, a few steps must be executed by the manufacturer. After they have finished developing a new firmware version, they first compute the hash of the firmware file. Although a digital signature offers better security, the slightly less secure hash function SHA-256 was chosen for this system. The reason for this is the significantly shorter computation time of the hash on a constrained device compared to a digital signature. The use of this hash function provides sufficient collision resistance to guarantee authenticity and is also recommended by the german *Federal Office for Information Security (BSI)* and the american *National Institute of Standards and Technology (NIST)* for being utilised as a hash function [15, 16]. Subsequently, an unencrypted firmware image is created using the computed hash and the firmware file. Through the symmetrical encryption method AES-256 in CBC mode the confidentiality of the new firmware image is provided. The key required to encrypt the firmware image, also called the manufacturer key, is the same for every device and is hard-coded in the firmware of the bootloader. Since the hash is also encrypted, a similar level of security is achieved as by using a MAC. As a final step the resulting encrypted firmware image is uploaded to the firmware update server of the company. Figure 2 illustrates this process.

With the smartphone app, the user can subsequently check for available firmware updates. In case a pending update is available they can initiate the installation of the new firmware image. To make sure that confidentiality is kept during the transfer of the update from the smartphone app to the device, the symmetrical encryption method AES-256 is applied here as well. In contrast to the encryption of the firmware image at the place of the manufacturer each system uses a different base key which is stored on the device during production. The smartphone app obtains the key by scanning a QR code that the user receives when purchasing the system. This key is used to derive a message key for each individual data transmission. To maintain integrity each transmitted message contains a CRC value and is acknowledged by the receiver. This allows errors to be detected and prevents packet loss. An additional sequence

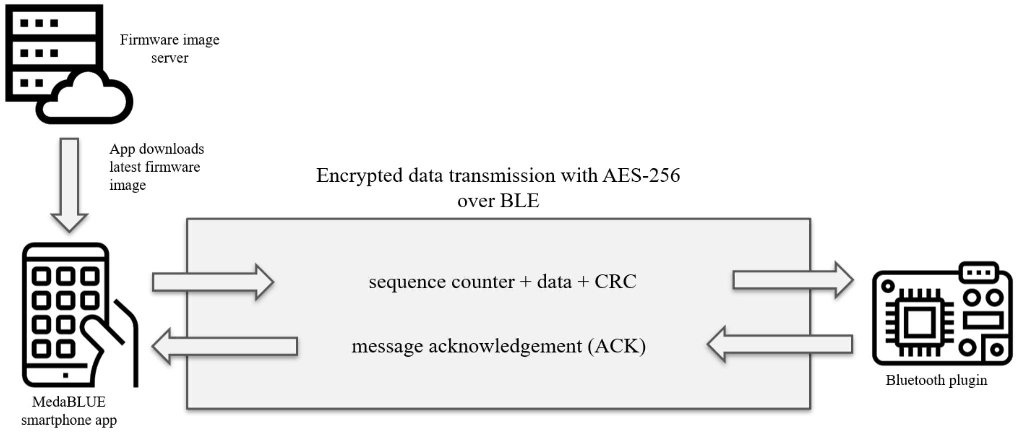

Firmware image server
App downloads latest firmware image
Encrypted data transmission with AES-256 over BLE
sequence counter + data + CRC
message acknowledgement (ACK)
MedaBLUE smartphone app
Bluetooth plugin

Figure 3: Transmission process of a firmware image from the app to the embedded system.

counter also preserves the integrity of the communication over the default maximum transmission unit of the bluetooth protocol. This allows a message to be sent that is larger than the default Maximum Transmission Unit (MTU) of 23 bytes [17].

Thus, this firmware update meets the security concepts mentioned above and enables a secure firmware update from the smartphone app. In addition to compliance with the security concepts, protection against tampering is also provided by deactivating the debug interface during production. This means that access to sensitive data in the system memory is no longer possible. Figure 3 summarizes the transmission process of a new firmware image.

To maintain all the systems tasks during the transmission of the firmware image the application stores all data of the received firmware image in an external flash memory. The basic memory layout of the external memory and the use of the individual sections are described in chapter 3.1 and Figure 1. The use of the external memory described there guarantees a fallback solution in the case of an error, as the stable running firmware is always stored in the memory.

4.2 Update Execution in the Bootloader

After successfully transmitting the new firmware image, the embedded system starts the bootloader. Figure 4 shows the sequence of the firmware update in the system, which is similar to the one proposed by Kvarda *et al.* [5]. Assuming a new firmware is available, the bootloader first decrypts the firmware image with the manufacturer key and calculates the hash of the firmware file. A correct hash results in updating the firmware. In this step the bootloader writes the new firmware into the internal flash memory of the microcontroller and sets the status of the firmware to *Not Running*. An incorrect hash leads to the new firmware being ignored and the currently installed application being started. This is also done if the new firmware was successfull installed on the device.

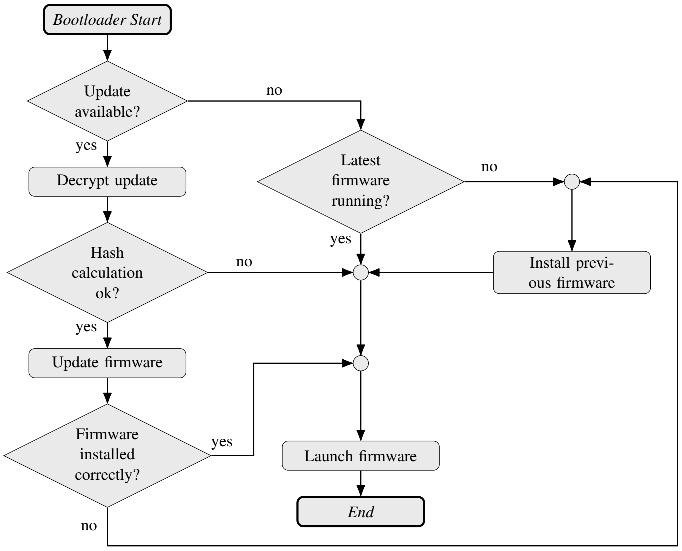


Figure 4: Process of a secure firmware update performed by the bootloader of the system.

After initially entering the application of the new firmware the system self-checks the running state of the firmware. Therefore the application initializes its system and verifies whether the communication to its peripheral components is possible. If the system can further communicate with the connected digital electricity meter and start a Bluetooth broadcast, the basic function of the system is given. With this, any case of an error occurrence or dysfunction regarding the stability or the communication of the new firmware image can not only be detected but also resolved. When an error is detected during this process the system starts the bootloader again. It then checks whether the latest installed firmware is running. As this is not the case the bootloader automatically downgrades the application by installing the previous firmware stored in the other section of the external flash memory (Figure 1). If there is no error regarding the new firmware, the status is set to *Current Firmware* whereas the previous firmware is set to *Previous Firmware*. The memory section marked in such a way will be overwritten during the next update. With each new firmware update, this process switches between the two memory areas shown in Figure 1, which are identified only by the status variables. To prevent the exploitation of a bug of previous firmware versions, a downgrade is only allowed to the previous firmware that is already stored in the external memory. This mechanism guarantees that the system is capable to select the correct firmware image in the external memory and is always running with a stable and secure firmware.

5 Conclusion

This paper described challenges and solutions for secure firmware updates. The aim was to show the possibilities to perform firmware updates on embedded systems securely by the user. Thereby, the compliance with the security concepts is essential to ensure that system attacks during a firmware update remain less interesting due to the high time and financial effort. Nevertheless, for each system it is necessary to weigh which methods are necessary for the security of the system based on the purpose of the application. Despite all the security concepts presented, vulnerabilities may still remain if users are not motivated to update the firmware of their own devices. Using all methods is not recommended due to the numerous computationally and resource-intensive operations. This would also be associated with high development costs.

In addition, a possible sequence of a secure update process on an embedded system was presented. This offers a balanced solution between computing effort and security. The proposed secure software update process is particularly interesting for systems with low computing power. In future works, the software process will be further developed and improved with regard to the performance and security of the system.

Acknowledgment

We gratefully acknowledge the support of DZG Metering GmbH, which have provided an insight to their secure firmware update process.

References

[1] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, “DDoS in the IoT: Mirai and Other Botnets”, *Computer*, vol. 50, no. 7, pp. 80–84, 2017. DOI: 10.1109/MC.2017.201.

[2] M. Antonakakis, T. April, M. Bailey, *et al.*, “Understanding the Mirai Botnet”, in *26th USENIX Security Symposium (USENIX Security 17)*, 2017, pp. 1093–1110. [Online]. Available at: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis.

[3] B. Krebs, “New Mirai Worm Knocks 900K Germans Offline”, Krebs on Security, Ed., 2016. [Online]. Available at: https://krebsonsecurity.com/2016/11/new-mirai-worm-knocks-900k-germans-offline/.

[4] A. Kolehmainen, “Secure Firmware Updates for IoT: A Survey”, in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, IEEE, 2018, pp. 112–117. DOI: `10.1109/Cybermatics_2018.2018.00051`.

[5] L. Kvarda, P. Hnyk, L. Vojtech, and M. Neruda, “Software Implementation of Secure Firmware Update in IoT Concept”, *Advances in Electrical and Electronic Engineering*, vol. 15, no. 4, 2017. DOI: `10.15598/aeee.v15i4.2467`.

[6] S. Falas, C. Konstantinou, and M. K. Michael, “A Modular End-to-End Framework for Secure Firmware Updates on Embedded Systems”, 2020. [Online]. Available at: `https://arxiv.org/pdf/2007.09071`.

[7] Texas Instruments Incorporated, “Secure In-Field Firmware Updates for MSP MCUs”, Texas Instruments Incorporated, Ed., 2015. [Online]. Available at: `https://www.ti.com/lit/an/slaa682/slaa682.pdf`.

[8] N. S. Mtetwa, P. Tarwireyi, A. M. Abu-Mahfouz, and M. O. Adigun, “Secure Firmware Updates in the Internet of Things: A survey”, in *2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, Piscataway, NJ: IEEE, 2019, pp. 1–7. DOI: `10.1109/IMITEC45504.2019.9015845`.

[9] C. Eckert, “IT-Sicherheit: Konzepte - Verfahren - Protokolle”, 10th ed. Munich: De Gruyter Oldenbourg, 2018. DOI: `10.1515/9783110563900`.

[10] L. Keleman, D. Matic, M. Popovic, and I. Kastelan, “Secure firmware update in embedded systems”, in *Proceedings 2019 IEEE 9th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, G. Velikić, Ed., Piscataway, NJ: IEEE, 2019, pp. 16–19. DOI: `10.1109/ICCE-Berlin47944.2019.8966174`.

[11] Atmel Corporation, “Safe and Secure Bootloader Implementation for SAM3/4: Atmel 32-bit Microcontroller”, Atmel Corporation, Ed., 2013. [Online]. Available at: `http://ww1.microchip.com/downloads/en/Appnotes/Atmel-42141-SAM-AT02333-Safe-and-Secure-Bootloader-Implementation-for-SAM3-4_Application-Note.pdf`.

[12] K. Zandberg, K. Schleiser, F. Acosta, H. Tschofenig, and E. Baccelli, “Secure Firmware Updates for Constrained IoT Devices Using Open Standards: A Reality Check”, *IEEE Access*, vol. 7, pp. 71 907–71 920, 2019. DOI: `10.1109/ACCESS.2019.2919760`.

[13] DZG Metering GmbH, “MEDAblue”, 2020. [Online]. Available at: `https://play.google.com/store/apps/details?id=com.DZG.MEDAblue&gl=DE`.

[14] ——, “MEDAblue”, 2020. [Online]. Available at: `https://apps.apple.com/de/app/meda-blue/id1487362534`.

[15] Bundesamt für Sicherheit in der Informationstechnik, “BSI TR-02102 Kryptographische Verfahren: Empfehlungen und Schlüssellängen”, 2021. [Online]. Available at: `https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.pdf`.

[16] National Institute of Standards and Technology, “Secure Hash Standard (SHS)”, Washington, D.C., 2015. DOI: `10.6028/NIST.FIPS.180-4`.

[17] Bluetooth SIG, “Bluetooth Core Specification 5.2”. [Online]. Available at: `https://www.bluetooth.com/specifications/specs/core-specification/`.

Unconscious Check-in : An Empirical Analysis of OAuth CSRF in the Wild

Michele Benolli, Seyed Ali Mirheidari, Elham Arshad, Bruno Crispo
Uinversity of Trento, Italy

Keywords. OAuth; Request Forgery; CSRF; Login CSRF

Abstract

OAuth 2.0 is a popular and industry-standard protocol. To date, different attack classes and relevant countermeasures have been proposed. However, despite the presence of guidelines and best practices, the current implementations are still vulnerable and error-prone. This research mainly focused on Cross-Site Request Forgery (CSRF) attack. This attack is one of the dangerous vulnerabilities in OAuth protocol, which has been mitigated through `state` parameter. However, despite the presence of this parameter in the OAuth deployment, many websites are still vulnerable to OAuth-CSRF (OCSRF) attack.

We studied one of the most recurrent type of OCSRF attack through a variety range of novel attack strategies based on different possible implementation weaknesses and the state of the victimâs browser at the time of the attack. In order to validate them, we designed a repeatable methodology and conducted a large-scale analysis on 395 high-ranked sites to assess the prevalence of OCSRF vulnerabilities. Our automated crawler discovered about 36% of targeted sites are still vulnerable and detected about 20% more well-hidden vulnerable sites utilizing the novel attack strategies.

Based on our experiment, there was a significant rise in the number of OCSRF protection compared to the past scale analyses and yet over 25% of sites are exploitable to at least one proposed attack strategy. Despite a standard countermeasure exists to mitigate the OCSRF, our study shows that lack of awareness about implementation mistakes is an important reason for a significant number of vulnerable sites.

References

[1] Avinash Sudhodanan, Roberto Carbone, Luca Compagna, Nicolas Dolgin, Alessan-dro Armando, and Umberto Morelli. 2017. Large-scale analysis detection of authentication cross-site request forgeries. In 2017 IEEEE uropean symposiumon security and privacy(EuroSP). IEEE, 350â365.

[2] Adam Barth, Collin Jackson, and John C Mitchell. 2008. Robust defenses for cross-site request forgery. In Proceedings of the 15th ACM conference on Computer and communications security. 75â88.

[3] San-Tsai Sun and Konstantin Beznosov. 2012. The devil is in the (implementation)details: an empirical analysis of OAuth SSO systems. In Proceedings of the 2012 ACM conference on Computer and communications security. 378â390.

[4] Karin Sumongkayothin, Pakpoom Rachtrachoo, Arnuphap Yupuech, and Kasidit Siriporn. 2019. OVERSCAN: OAuth 2.0 Scanner for Missing Parameters. In International Conference on Network and System Security. Springer, 221â233.

[5] Wanpeng Li, Chris J Mitchell, and Thomas Chen. 2019. OAuthGuard: Protecting User Security and Privacy with OAuth 2.0 and OpenID Connect. In Proceedings of the 5th ACM Workshop on Security Standardisation Research Workshop. 35â44.

Standard Cell Design For Data-Independent Static Power Under Illumination

Jan Bělohoubek, Petr Fišer and Jan Schmidt
Czech Technical University in Prague
Prague, Czech Republic

{jan.belohoubek, petr.fiser, jan.schmidt}@fit.cvut.cz

Keywords. Static Power, Subthreshold Leakage, Optical Beam Induced Current (OBIC), CMOS, Standard Cell, SPICE, AES SBOX

Abstract

Physical attacks, namely invasive, observation, and combined, represent a great challenge for today's digital design. Successful class of strategies adopted by industry, allowing hiding data dependency of the side channel emissions in CMOS is based on balancing. Although attacks on CMOS dynamic power represent a class of state-of-the-art attacks, vulnerabilities exploiting data dependency in CMOS static power [1, 2, 3, 4] and light-modulated static power were recently presented [5, 6, 7].

In this contribution are presented the structures and techniques developed to enhance and balance the power imprint of the traditional static CMOS bulk structures under invasive light attack. The idea behind the balancing is to mimic the behavior of balanced inverter chains (see Figure 1) in a more complex CMOS cell. The microarchitecture of the proposed cells is inspired by the connection approximating the *Constant Current Source*.

The standard cell design was confirmed by SPICE simulations using the validated SPICE models for CMOS under *Photoelectric Laser Stimulation* (PLS) [8, 9].

The novel standard cells designed according to the presented techniques in the TSMC180nm technology node were used to synthesize the dual-rail AES SBOX block. The behavior of the AES SBOX block composed of the novel cells is compared to classical approaches. Usage of novel cells enhances circuit security under invasive light attacks while preserving comparable circuit resistance against state-of-the-art power attacks.

The models and resources used for this research are available online [10].

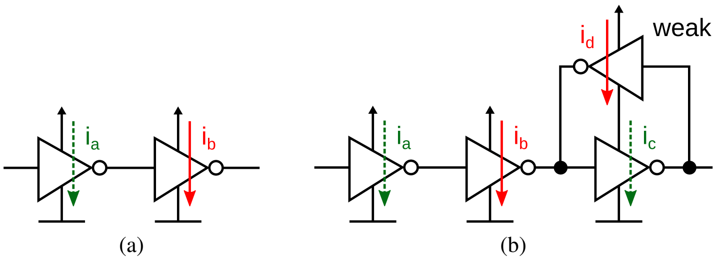


Figure 1: Two-inverter chain (a) uses complementary power consumption to obtain a constant power imprint: $i_a + i_b = const.$; three-inverter chain with feedback weak inverter (b) uses the same principle

Paper origin

This work has been accepted and presented at the 23rd International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS 2020), while the extended version of the original contribution is currently under review in the Microelectronics Reliability journal.

Acknowledgment

The authors acknowledge the support of the OP VVV MEYS funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics". Computational resources were supplied by the project "e-Infrastruktura CZ" (e-INFRA LM2018140) provided within the program Projects of Large Research, Development and Innovations Infrastructures. The novel structures presented in this paper are subject of the patent application.

References

[1] J. Giorgetti, G. Scotti, A. Simonetti, and A. Trifiletti, "Analysis of data dependence of leakage current in CMOS cryptographic hardware," in *Proceedings of the 17th ACM Great Lakes symposium on VLSI.* ACM, 2007, pp. 78–83.

[2] M. Alioto, L. Giancane, G. Scotti, and A. Trifiletti, "Leakage Power Analysis attacks: Well-defined procedure and first experimental results," in *2009 International Conference on Microelectronics - ICM*, Dec 2009, pp. 46–49.

[3] T. Moos, A. Moradi, and B. Richter, "Static Power Side-Channel Analysis – An Investigation of Measurement Factors," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2019.

[4] B. Fadaeinia, T. Moos, and A. Moradi, "BSPL: Balanced Static Power Logic." *IACR Cryptol. ePrint Arch.*, vol. 2020, p. 558, 2020.

[5] J. Bělohoubek, P. Fišer, and J. Schmidt, "Using Voters May Lead to Secret Leakage," in *IEEE 22nd International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS 2019)*, April 2019, pp. 1–4.

[6] ——, "CMOS Illumination Discloses Processed Data," in *22nd Euromicro Conference on Digital System Design (DSD 2019)*, Aug 2019, pp. 381–388.

[7] ——, "Standard Cell Tuning Enables Data-Independent Static Power Consumption," in *23rd International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS 2020)*, Apr 2020.

[8] A. Sarafianos, R. Llido, O. Gagliano, V. Serradeil, M. Lisart *et al.*, "Building the electrical model of the pulsed photoelectric laser stimulation of an NMOS transistor in 90nm technology," in *38th International Symposium for Testing and Failure Analysis, (ISTFA) 2012*, 2012, pp. 5B–5.

[9] A. Sarafianos, O. Gagliano, M. Lisart, V. Serradeil, J. Dutertre, and A. Tria, "Building the electrical model of the pulsed photoelectric laser stimulation of a PMOS transistor in 90nm technology," in *Proceedings of the 20th IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*, July 2013, pp. 22–27.

[10] J. Bělohoubek. (2019 – 2021) Photoelectric Laser Stimulation of Combinational Logic. [Online]. Available: https://github.com/DDD-FIT-CTU/CMOS-PLS/

Fast fixed-point arithmetic in three-valued binary number abstraction

Jan Onderka
Faculty of Information Technology
Czech Technical University in Prague
Prague, Czech Republic

onderjan@fit.cvut.cz

Keywords. formal verification, model checking, three-valued abstraction, unknown value, addition, subtraction, multiplication, ripple-carry adder, polynomial time, pseudoboolean function, modular inequality

Abstract

The paper presents novel methods and fast algorithms for performing addition, subtraction and multiplication in three-valued abstraction used in finite-state formal verification, where bits are represented by values zero, one, and unknown (potentially zero, potentially one). The forward problem is considered, in which a single abstract bit vector result is generated from given abstract bit vector operands. The result must contain all possible concrete results to avoid underapproximation, which would lead to unsound verification. It is also desirable that the found abstract result is the best one, i.e. contains the smallest possible number of concrete results to minimize overapproximation.

The problem of finding the best abstract result can be trivially solved in time exponential to the number of input bits for any operation. This is infeasible for real-world adders and multipliers, however, and a worst-case polynomial-time algorithm is desired. In the paper, it is shown that propagating the minimum and maximum ripple-carry adders carryovers results in a linear-time algorithm finding the best abstract result for addition and subtraction.

Such an algorithm does not result in the best result for multiplication, however. The problem is reformulated via pseudoboolean functions to a set of quantified modular inequalities. A novel approach using global extreme-finding with maximum-step guarantees is devised to solve them and a worst-case quadratic-time algorithm is obtained and proven to be produce the best result.

Paper origin

This paper has not been published yet and is planned to be published in the near future.

Acknowledgment

This work was supported by the Czech Technical University (CTU) grant No. SGS20/211/OHK3/3T/18.

Self-Test Libraries Analysis for Pipelined Processors Transition Fault Coverage Improvement

Riccardo Cantoro[1], Patrick Girard[2], Riccardo Masante[1], Sandro Sartoni[1], Matteo Sonza Reorda[1], Arnaud Virazel[2]

[1]Politecnico di Torino
Turin, Italy

[2]LIRMM
University of Montpellier / CNRS
Montpellier, France

Keywords. software-based self-test, software test libraries, on-line test, transition delay test, safety, functional test

Abstract

Testing digital integrated circuits is generally done using Design-for-Testability (DfT) solutions. Such solutions, however, introduce non-negligible area and timing overheads that can be overcome by adopting functional solutions. In particular, functional test of integrated circuits plays a key role when guaranteeing the device's safety is required during the operative lifetime (*in-field test*), as required by standards like ISO26262. This can be achieved via the execution of a Self-Test Library (STL) by the device under test (DUT). Nevertheless, developing such test programs requires a significant manual effort, and can be non-trivial when dealing with complex modules. This paper moves the first step in defining a generic and systematic methodology to improve transition delay faults' observability of existing STLs. To do so, we analyze previously devised STLs in order to highlight specific points within test programs to be improved, leading to an increase in the final fault coverage.

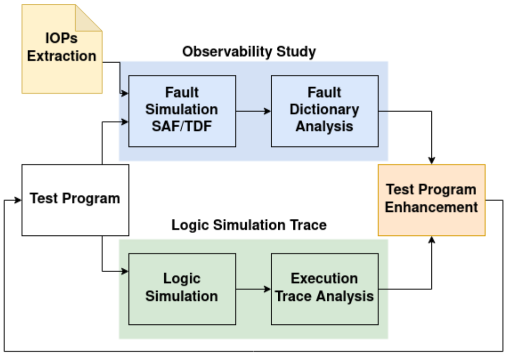


Figure 1: Proposed test flow

1.1 Proposed Approach

Fig. 1 shows the proposed methodology, which can be divided into two different processes:

1. **Observability study:** this process aims to give some insights on not observed (NO) faults: devising test strategies for such faults depends on where their effects propagated and stopped. For this reason, we define two groups of internal observation points (IOPs), namely *User Accessible Registers (UARs)*, registers directly accessible by the user through available instructions, and *Hidden Registers (HRs)*, hidden within sub-modules or glue logic and not directly accessible. In this process, we generate a fault dictionary, which includes timing information on fault detection. Moreover, we analyze and correlate stuck-at fault (SAF) and transition delay fault (TDF) coverages, based on the implications existing between the two fault models, as testing a TDF implies testing the relative SAF.

2. **Logic Simulation Trace:** this process allows to map the execution time to the instructions currently executed by the processor core.

Combining data from the fault dictionary and the execution trace, the proposed test flow allows to easily identify what portion of the code must be improved to cover NO faults. Which and how many instructions to use in general depends on the IOP reached by the fault and is not the main focus of the current work.

1.2 Experimental results

The approach presented in this work has been validated on the open-source SoC PULPino [1], which has been synthesized using a 45nm library and resulted in a total number 159, 326 transition delay faults. As for the test programs, we used three STLs based on different algorithms to test stuck-at faults. Data on these STLs, together with the experimental results, are reported in Table 1.

Table 1: STLs general information

Test Program	*#Clock cycles*	*Memory size [kB]*	*SAF coverage %*	*Initial TDF coverage %*	*Potential TDF coverage %*	*Potential gain [percentile units]*
STL1	17, 308	27.32	81.42	61.73	70.88	9.15
STL2	31, 158	27.86	81.86	44.19	53.15	8.96
STL3	80, 455	16.68	82.18	62.54	80.39	17.85

Results show that, assuming the test engineer can deploy strategies to recover all NO faults, a total number of 14, 580, 14, 275, and 28, 433 transition delay fault can be recovered for STL1, STL2, and STL3 respectively, leading to the potential gain increment reported in the results table.

Future works will focus on the development of strategies to observe effects of NO faults at the DUT's primary outputs.

Paper origin

This paper has been accepted and presented at the conference IEEE International Symposium on On-Line Testing and Robust System Design (IOLTS) 2021.

References

[1] ETH Zurich and Università di Bologna: PULPino microcontroller system, 2020, available online: *https://github.com/pulp-platform/pulpino*

Brief Summary of Observations and Research on Encrypted DNS

Dmitrii Vekshin[*†‡], **Karel Hynek**[†‡], **Tomáš Čejka**[†‡]
* Author is with: Avast s.r.o., Pikrtova 1737, Prague 4,
† Author is with: FIT CTU in Prague, Thákurova 9 Prague 6,
‡ Author is with: CESNET z.s.p.o., Zikova 4, Prague 6

vekshdmi@fit.cvut.cz, hynekkar@fit.cvut.cz, cejkato2@fit.cvut.cz

Keywords. DNS over HTTPS, DoH, Detection, Classification, Machine Learning, Datasets

Abstract

Encrypted traffic increases privacy levels for users; however, it also hides information from monitoring and security systems. Recently, several newly defined protocols bring encryption for domain name translation mechanisms as well. Migration to Encrypted DNS becomes transparent for users since modern software enables this technology by default. As a side effect, various security threats exploit encryption by hiding malicious activity in the traffic. The goal of our research is to study the privacy and security of this area and explore possibilities of automatic classification and detection based on extended IP flows. This paper is a summary of recently published or submitted works of our team.

1 Introduction

The DNS over HTTPS (DoH) protocol has been recently developed to remediate the privacy issues of the Domain Name System (DNS) protocol. DNS transmits queries in plain text, and these queries can reveal sensitive information like a user's browsing habits. The primary motivation for DoH is to limit the users' surveillance and protect them from possible profiling of their activities. Despite the fact that the protocol specification was published in 2018, it has already spread vastly. Most modern web browsers and operating systems already support DoH, and some of them enabled it by default. Besides DoH, there are several similar protocols, which allow hiding domain name translation. During our research, we analyzed DNS over TLS (DoT) and DNS over QUIC (DoQ) — altogether called Encrypted DNS.

All Encrypted DNS protocols work similarly. The HTTPS/TLS/QUIC encrypted connection is established from a client endpoint to the Encrypted DNS server to perform domain name resolution. After the connection is established, the client can resolve the required domain name in an encrypted manner. Even though it seems like a good way to enhance privacy, it can be a crucial issue to security. The reason is that after encrypting, the quires became invisible to network monitoring and detection systems. Consequently, it brings new risks to network administrate and helps malware actors.

2 Goals

Our motivation is to find the border between the privacy and security sides of Encrypted DNS protocols. Or even find a trade-off between them. Our main apprehension is that Encrypted DNS protocol does not guarantee privacy, but only changes decentralized DNS system to a centralized Encrypted DNS system, thereby forces users to trust the Encryption DNS provider. Also, we observed increasing use of Encrypted DNS in unwanted traffic like malware and bypassing security policies. Therefore, we aim to study possibilities of automatic detection and classification of Encrypted DNS traffic and observation of dissemination of this technology.

3 Our research

In the initial phase of our research, we faced a problem that the DoH protocol is hard to block since it shares 443 TCP port with other HTTPS traffic. Therefore, we experimented with a new machine learning model that achieved very high accuracy in recognition of DoH and Not DoH traffic in (2).

Furthermore, the paper analyzed possible leaks of private information, even when the domain name resolution process is encrypted. We researched the behavior of the DoH protocol implementation in web browsers and the level of detail that can be revealed by observing and analyzing packet-level information. The aim was to evaluate and highlight discovered privacy weaknesses hidden in DoH. This research continued in (1), where the trained machine learning classifier was able to infer individual domain names based on the DoH connections. The presented classifier achieved surprisingly high accuracy up to 90 % on HTTP 1.1, and up to 70 % on HTTP 2 protocol.

According to our observations during the research of encrypting DNS queries using DoH entails security threats. We analyzed malicious and unwanted activities that leverage DoH and can be currently observed in the wild. We identified three real-world abuse scenarios discovered in the web environment that reveal how service providers intentionally use DoH to violate policies. The most significant examples of abuse of DoH protocol on the web are observed in the illegal gambling industry, pirate content distribution. Moreover, we found malware samples that weaponize DoH protocol to hide their activities on the network. This topic was described and submitted to the Computers&Security journal.

Recently, we finished our latest research on Encrypted DNS prevalence. In this research, we observed five months of traffic at the beginning of 2021 by three different organizations with global coverage. By comparing the total values of the number of requests per user, and the traffic seasonality, it was possible to infer current adoption trends. Moreover, we actively scanned the whole Internet for still-unknown working DoH servers, and we compared them with a novel curated list of well-known DoH servers. We conclude there was statistically significant evidence that the average amount of Internet traffic for DoH, DoT, and DoQ remained stationary. However, we found that the number of DoH servers increased 4 times. This research was described in detail in the paper submitted to IMC2021.

4 Conclusion

In the field of Encrypted DNS protocols, there are many research challenges, especially regarding automatic detection and blocking of malicious traffic. This paper described some recent works exploring this topic. According to our observations, the volume of Encrypted DNS increases, and there are already known malware samples that use it. Therefore, this research is essential to minimize security risks.

References

[1] Hynek, K., Cejka, T.: Privacy illusion: Beware of unpadded doh. In: 2020 11th IEEE Annual IEMCON. pp. 0621–0628 (2020). https://doi.org/10.1109/IEMCON51383.2020.9284864

[2] Vekshin, D., Hynek, K., Cejka, T.: Doh insight: Detecting dns over https by machine learning. In: Proceedings of the 15th International Conference on Availability, Reliability and Security. ARES ’20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3407023.3409192

System of automatic approach control for a tractor unit's approach to a semi-trailer

Oskar Jędrzejczak, Piotr Mróz, Małgorzata Mazurkiewicz
Affiliation (Faculty of Informatics, Electrotechnics And Automatics,
University of Zielona Gora)
Prof. Z. Szafrana Str. 2
65 - 516 Zielona Góra
97235@g.wiea.uz.zgora.pl,
p.mroz@imei.uz.zgora.pl,
m.mazurkiewicz@issi.uz.zgora.pl

Abstract: Intelligent transportation systems are one of very quickly developing branches of industry. Their main goals include increasing transportation effectiveness, improving safety on the roads, recognizing vehicles, recognizing license plates, and supporting the driver by, for example, performing the function of automatic parking. The article presents a system performing the function of autonomous approach of a tractor unit to a semi-trailer. Among others, image recognition was used to realize the system. Based on the Raspberry Pi microcomputer, a subsystem was developed which, on the basis of the coordinates of the middle of the semi-trailer in regard to the middle of the tractor unit, allows for autonomous approach of the tractor unit to the semi-trailer. To indicate the middle of the semi-trailer, a camera placed in the backside of the tractor unit's cabin was used. The system works correctly and is used in models of tractor units with semi-trailers at the Faculty of Informatics, Electrotechnics and Automatics, University of Zielona Gora. The experiences gained are one of the stages of building a system supporting the approach to the semi-trailer and automatic regulation of the height of the fifth-wheel coupling in a real tractor unit.

Keywords. Intelligent transportation systems, control systems, image recognition, control algorithms.

1. Introduction

Currently, intelligent transportation systems ("ITS") are a very quickly developing field of technology. These systems include solutions utilizing various technologies: information technology, telecommunication, data processing, as well as infrastructure, vehicles and their users. [1, 2, 3]. Among the most popular fields within ITS, we can include automatic movement of vehicles, tracking of the flow of movement in real time, vehicle and license plate recognition [4, 5, 6] or supporting automatic parking.

Driving large-size vehicles requires a high degree of skill from the driver, especially when it comes to safe and collision-free attachment of a tractor unit to a semi-trailer. The process of a tractor unit's approach to a semi-trailer is not an easy one. From the driver, it demands both speed and precision in realizing the task given to him.

In the case of heavy goods vehicles ("HGVs"), maneuvers such as reversing, parking or even loading of goods are a real challenge. They require concurrent operation of many factors, such as regulation of speed and wheel angle or observing the space around the vehicle. This is especially problematic in the case of reversing and approaching the semi-trailer, as elements of the tractor unit and the semi-trailer that must link are not visible to the driver. The precision of the performed maneuver is also affected by outside factors, such as weather conditions, lighting, or space available at the parking lot.

Very often, even experienced drivers have with precisely performing the maneuver of approaching the semi-trailer with the tractor unit, which can cause the car to crash into the semi-trailer, which often enough contains delicate items or even live animals.

The effects of such errors are tractor unit damage [7], (for example, the cracking of elements of the fifth-wheel coupling mechanism), semi-trailer damage, and damage to the goods being transported, which entails high vehicle repair costs and, often enough, payment of high compensation to the owners of the goods.

Due to this, a need to automate this process has developed. Automation would significantly improve the quality of life of HGV drivers and minimize the stress caused by maneuvers and the risk of mistakes.

The article presents a system performing the function of autonomous approach of a tractor unit to a semi-trailer. The system is based on image recognition. The images being processed are captured in real time by a camera installed on the backside of the tractor unit's cabin. A Raspberry Pi microcomputer was used to gather and analyze the images. As the system is realized with the use of, among others, the Raspberry Pi, due to how time-consuming the processing of image recognition algorithms is and how diverse the shapes of semi-trailers and lab conditions are, in the first phase of the tests, ArUco markers were used to recognize the type of semi-trailer and its axle. The system tests were conducted on models comprising of 4 different HGVs, which can drive with any semi-trailer (Figure. 1).

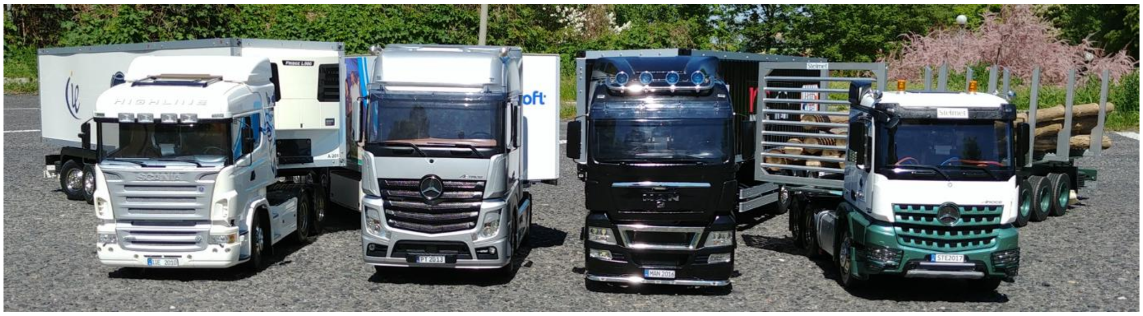


Figure 1. Models of tractor units with semi-trailers [8]

2. Structure of the system

Figure 2 shows a flowchart of the control system for a set comprised of a manipulator, a tractor unit and a semi-trailer. The set is controlled via a wireless ZigBee interface. The signal from the manipulator is sent to the tractor unit & performed there. The tractor unit returns a confirmation of order performance to the manipulator. At this time, an order responsible for controlling the semi-trailer is being prepared in the tractor unit. After this order is sent, the semi-trailer performs it and sends a confirmation of order performance to the tractor unit.

Figure 3 shows the structure of the control system [9]. It consists of an image gathering and processing subsystem and a tractor unit control subsystem. The tractor unit control subsystem communicates wirelessly with the manipulator and, on the basis of orders received, controls all functions of the vehicle and the semi-trailer. In HGV models, speed, direction of movement, gearbox, lights, and an MP3 player are controlled and actual speed and engine speed are measured. The subsystem is controlled by an STM32F409 processor.

Two cameras are attached to the image gathering and processing subsystem: a front one and a back one. The front camera is wide-angle. It is placed on the front side of the tractor unit cabinet. The second, regular camera is placed on the backside of the cabinet. It is this second camera that is the source of the signal which will be used for autonomous approach to the semi-trailer. Both sub-systems communicate with each other via a UART interface.

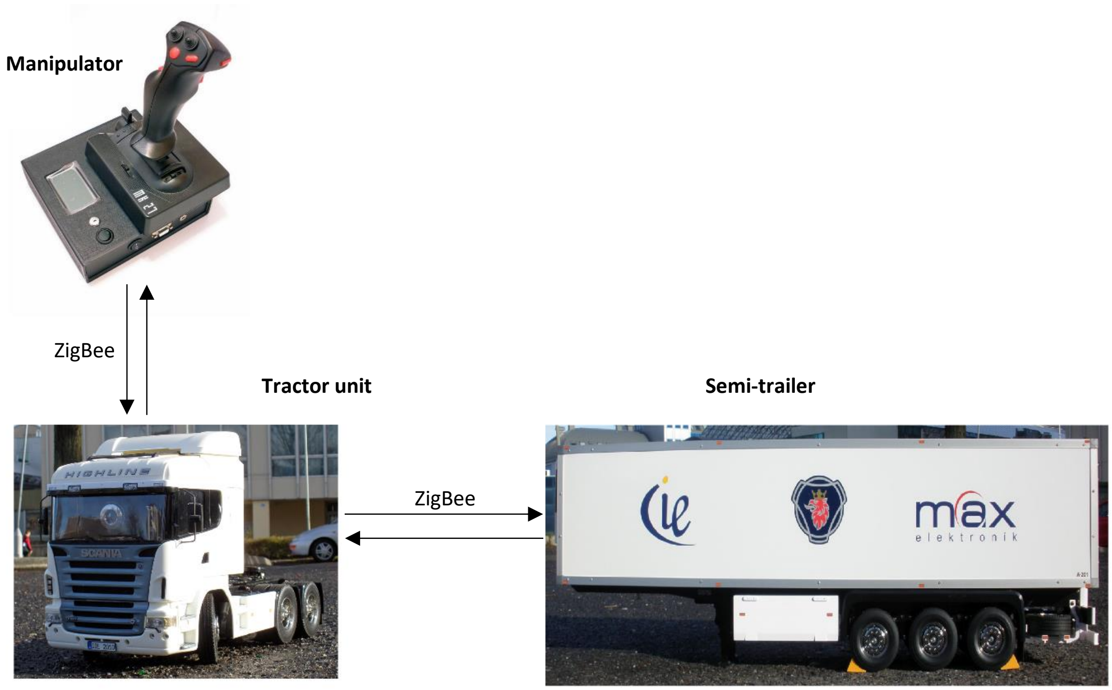


Figure 2. Flowchart of the control system for a set

3. The control algorithm

The results of the testing of the ArUco markers' fitness for the purpose of recognizing the type of semi-trailer were described in detail in article [10]. Before participating in the tests, each semi-trailer was marked with a printed image depicting the ArUco marker (Figure 4), which contained, among others, information about the type of semi-trailer.

Figure 5 shows the algorithm of the system's work with division of tasks between the image gathering and processing subsystem and the tractor unit control subsystem. After the control subsystem receives the order of autonomous approach, an order to begin the autonomous approach is sent to the image gathering and processing subsystem. Then, in the image gathering and processing subsystem, the process of identifying the type of semi-trailer starts. Due to how time-consuming the image recognition algorithms are and how diverse the shapes of real semi-trailers can be, a decision was made to use ArUco markers to recognize the type of semi-trailer and its axle.

Such a marker is placed on the axle of the semi-trailer, on its front side. The marker contains a unique semi-trailer code matching its type. During the identification of the semi-trailer type, the tractor unit control subsystem handles all orders from the manipulator. At this time, the person controlling the vehicle manually approaches the semi-trailer. Once the image gathering and processing subsystem identifies the type of semi-trailer, an order containing the type is sent to the tractor unit control subsystem. This is tantamount to identifying the semi-trailer type in the camera, without which the autonomous approach would be impossible. Receiving this order causes the vehicle to stop and interrupts the manual control from the manipulator. From this moment on, the tractor unit approaches automatically. The image gathering and processing subsystem determines the middle of the marker (which is also the middle of the semi-trailer) and generates the control vector on its basis. The control vector is comprised of the speed setting and the vehicle turn setting. The vector is sent to the HGV, where the settings are sent to executive subsystems. At the same time, in the tractor unit control subsystem, a check is performed to see if the semi-trailer has not appeared in the fifth-wheel coupling of the tractor unit. This information is sent to the image gathering and processing system. The algorithm covering the recognition of the semi-trailer axle, the determination of the control vector and the sending of the vector to the tractor unit is repeated until the semi-trailer is detected in the fifth-wheel coupling. Then, the vehicle is stopped, manual control from the manipulator is restored and the autonomous approach to the semi-trailer ends. The approach procedure can be stopped at any time via an order sent from the manipulator.

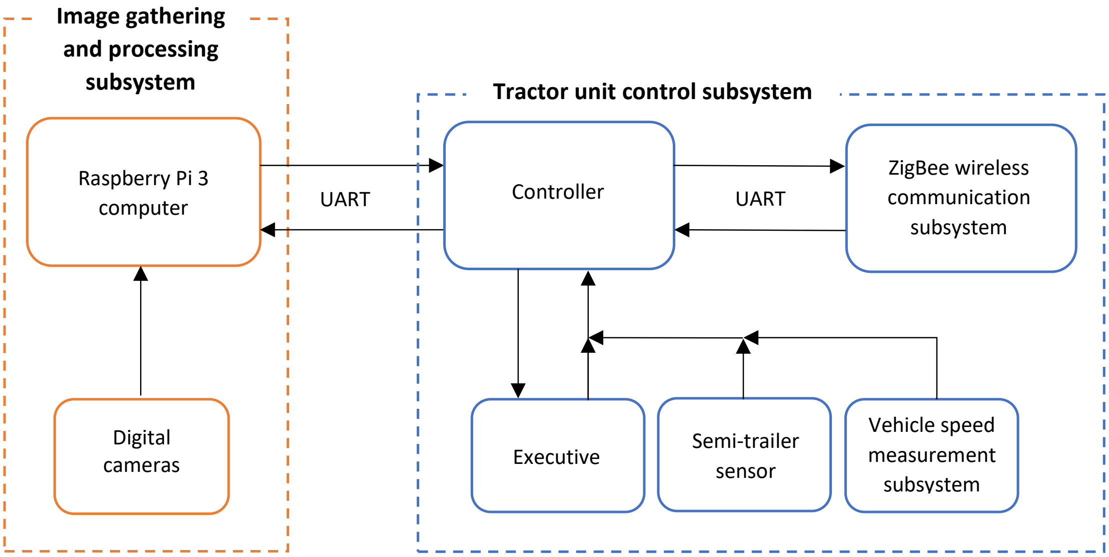


Image gathering and processing subsystem
Raspberry Pi 3 computer
Digital cameras
UART
Tractor unit control subsystem
Controller
UART
ZigBee wireless communication subsystem
Executive
Semi-trailer sensor
Vehicle speed measurement subsystem

Figure 3. The structure of the control system

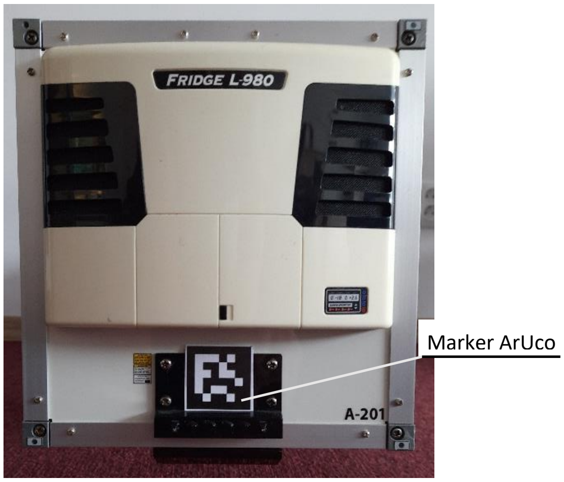


Figure 4. Semi-trailer marked with an ArUco marker

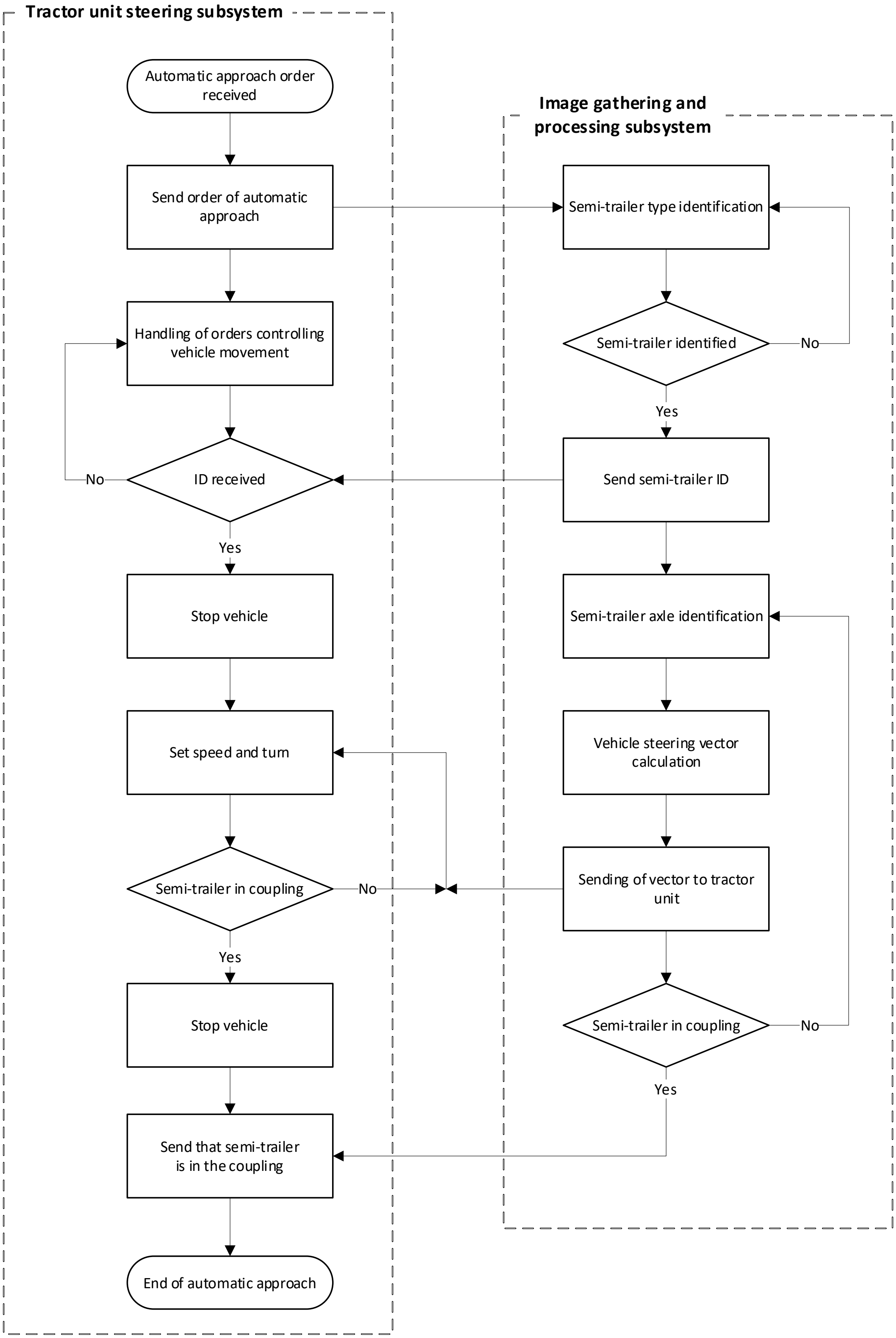
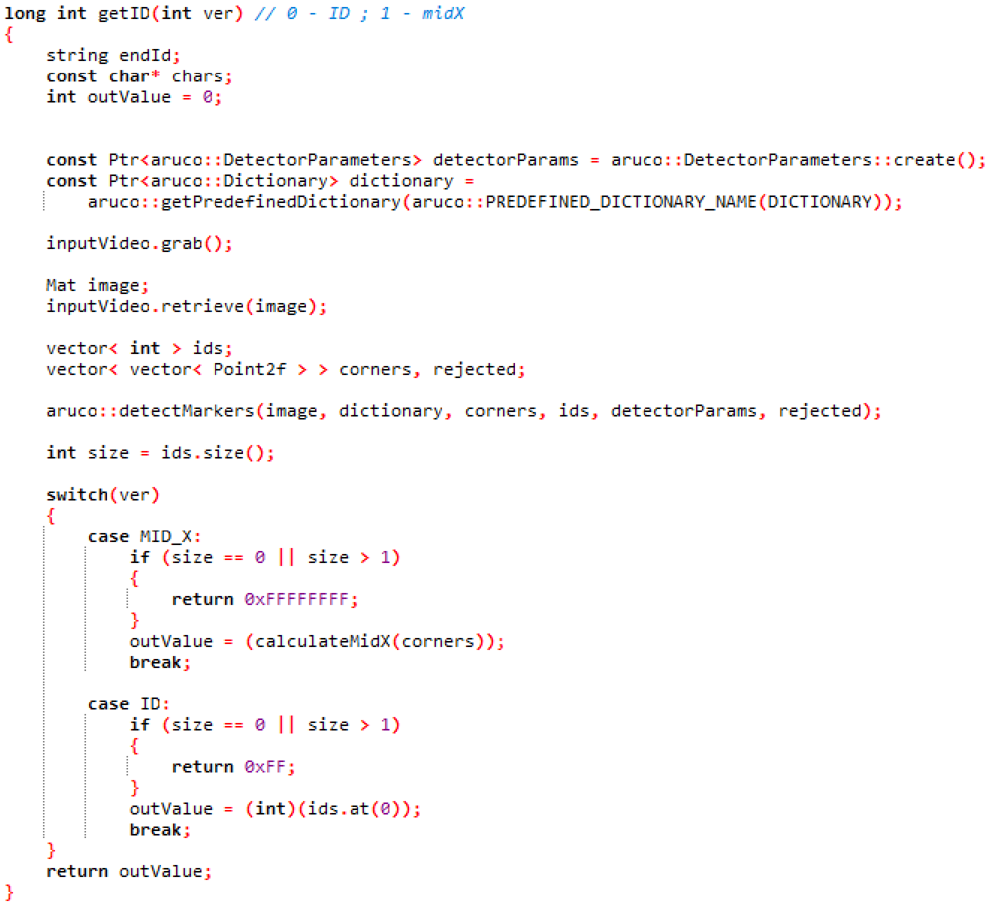

Tractor unit steering subsystem
Automatic approach order received
Send order of automatic approach
Handling of orders controlling vehicle movement
No
ID received
Yes
Stop vehicle
Set speed and turn
Semi-trailer in coupling
No
Yes
Stop vehicle
Send that semi-trailer is in the coupling
End of automatic approach
Image gathering and processing subsystem
Semi-trailer type identification
Semi-trailer identified
No
Yes
Send semi-trailer ID
Semi-trailer axle identification
Vehicle steering vector calculation
Sending of vector to tractor unit
Semi-trailer in coupling
No
Yes

Figure 5. The control algorithm

4. The image recognition algorithm

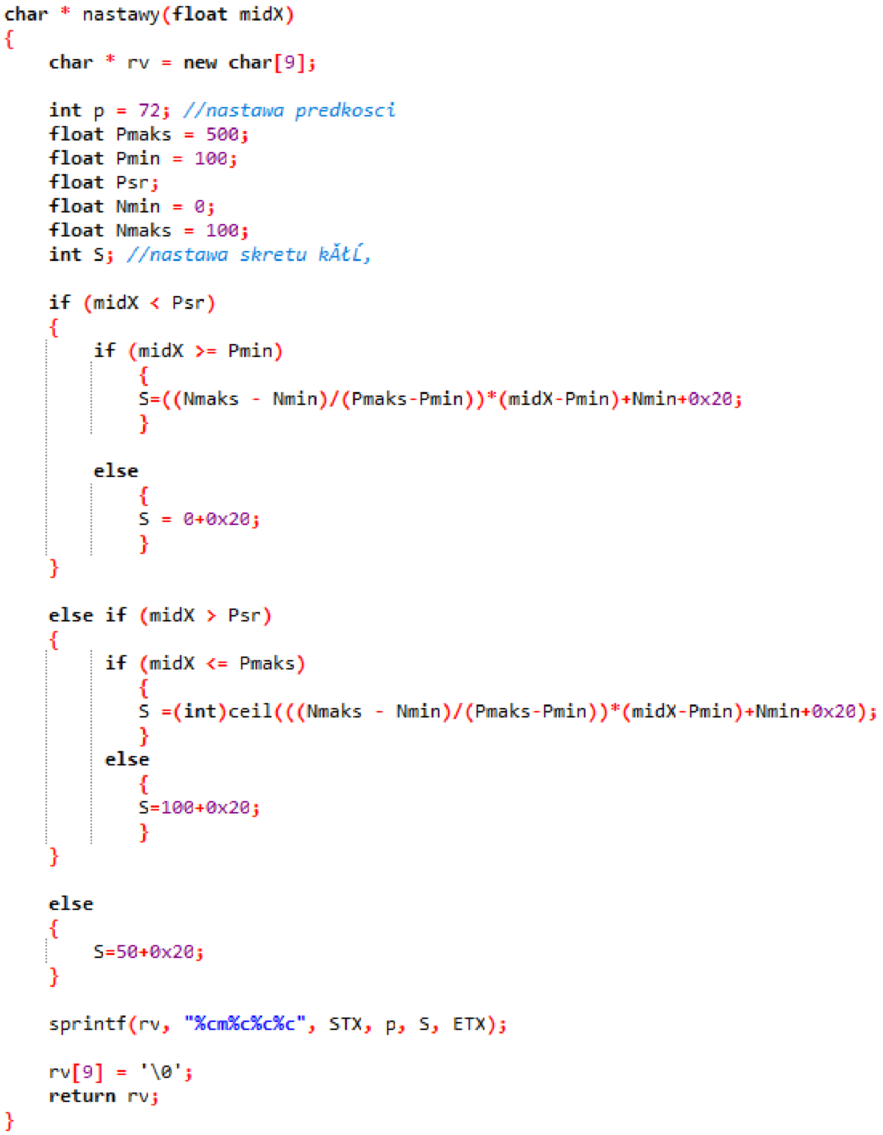
Figure 6 shows the code of the marker recognition function. The function is called with a parameter determining if the function should recognize the code imprinted on the marker or the coordinates of its middle. After the function is launched, variables and constants are defined and the image from the camera is captured. The next step is recognition of the ArUco marker within the image. Once the marker is recognized, a parameter value check occurs and an exit value is assigned from the function (either the ID of the marker or the coordinates of its middle).



```
long int getID(int ver) // 0 - ID ; 1 - midX
{
    string endId;
    const char* chars;
    int outValue = 0;


    const Ptr<aruco::DetectorParameters> detectorParams = aruco::DetectorParameters::create();
    const Ptr<aruco::Dictionary> dictionary =
        aruco::getPredefinedDictionary(aruco::PREDEFINED_DICTIONARY_NAME(DICTIONARY));

    inputVideo.grab();

    Mat image;
    inputVideo.retrieve(image);

    vector< int > ids;
    vector< vector< Point2f > > corners, rejected;

    aruco::detectMarkers(image, dictionary, corners, ids, detectorParams, rejected);

    int size = ids.size();

    switch(ver)
    {
        case MID_X:
            if (size == 0 || size > 1)
            {
                return 0xFFFFFFFF;
            }
            outValue = (calculateMidX(corners));
            break;

        case ID:
            if (size == 0 || size > 1)
            {
                return 0xFF;
            }
            outValue = (int)(ids.at(0));
            break;
    }
    return outValue;
}
```

Figure 6. Code of the marker recognition function

Figure 7 shows the control vector generation function. The control vector is comprised of two *unsigned char* type variables. The first, marked as *p*, is responsible for the speed setting, while the second, marked as *S*, is responsible for the turn setting. The setting values are set proportionally, on a range from 0 to 100. For the speed setting, a value of 0 means full speed in reverse, a value of 100 means full speed ahead, and a value of 50 means the vehicle stops. For the turn setting, a value of 0 means maximum turn to the right, a value of 100 means maximum turn to the left, and a value of 50 means that the wheels are straight. During the approach to the semi-trailer, the system automatically sets the lowest gear. The turn setting was chosen experimentally, so that the vehicle approaches reasonably quickly, but slowly enough to allow for generation of the turn setting. The determination of the turn setting starts with checking if the semi-trailer is to the left or to the right of the tractor unit. Then, depending on the drawback distance between the axle of the tractor unit and the axle of the semi-trailer, a turn setting is generated. If the distance exceeds the value determined in the program, the wheels are turned maximally to the left or to the right, depending on the position of the semi-trailer. Constants such as P_{maks} and P_{min}, which are part of the regulator determining the turn setting, are present in the algorithm. The values of these constants have an impact on the "accuracy" of the approach to the semi-trailer.



```
char * nastawy(float midX)
{
    char * rv = new char[9];

    int p = 72; //nastawa predkosci
    float Pmaks = 500;
    float Pmin = 100;
    float Psr;
    float Nmin = 0;
    float Nmaks = 100;
    int S; //nastawa skretu kĂtĹ,

    if (midX < Psr)
    {
        if (midX >= Pmin)
            {
            S=((Nmaks - Nmin)/(Pmaks-Pmin))*(midX-Pmin)+Nmin+0x20;
            }

        else
            {
            S = 0+0x20;
            }
    }

    else if (midX > Psr)
    {
         if (midX <= Pmaks)
            {
            S =(int)ceil(((Nmaks - Nmin)/(Pmaks-Pmin))*(midX-Pmin)+Nmin+0x20);
            }
         else
            {
            S=100+0x20;
            }
    }

    else
    {
        S=50+0x20;
    }

    sprintf(rv, "%cm%c%c%c", STX, p, S, ETX);

    rv[9] = '\0';
    return rv;
}
```

Figure 7. The contrrol vector generation function

For communication between the image gathering and processing subsystem and the tractor unit control subsystem, a UART interface is used. It has been assumed that signs controlling the transmission will be within the 0-1Fh range, while signs above this range will be used as data. Due to this, a value of 20h is added to the calculated value of the turn setting. In the case of the speed setting, this constant is already included earlier. Figure 8 shows the frame of the control vector prepared in the *printf* instruction. The beginning of data is marked with the STX sign (02h), while the end is marked with the ETX sign (03h).

The functions of image recognition and control vector generation are repeated until the tractor unit detects the semi-trailer's king pin in the unit's fifth-wheel coupling. At any time during the autonomous approach to the semi-trailer, if the operator notices a problem, he can send an order to the vehicle, stopping it and ending the approach.

start bit	speed setting	turn setting	stop bit
STX	*p*	*S*	ETX
02h	48h	*20h-84h*	03h

Figure 8. Frame of the control vector

5. Research results

The research was carried out in a lab. On the floor, a semi-trailer was placed, with a tractor unit placed some distance in front of it. An approach was started. At the beginning, the vehicle had issues with hitting the semi-trailer's king pin. After correcting the P_{maks} and P_{min} constants, the vehicle started approaching faultlessly. It has also turned out that it is beneficial to stop the vehicle when it identifies the semi-trailer ID, as from that moment on, the autonomous approach phase starts, and stopping the vehicle makes the start of the new phase clearly visible. Thanks to the ArUco markers, the semi-trailer axle recognition process ran smoothly and the whole approach to the semi-trailer was fluid.

6. Conclusion

The article presents an intelligent transportation system designed at the Faculty of Informatics, Electrotechnics and Automatics, University of Zielona Gora. The system is intended for the automation of the process of a tractor unit's approach to a semi-trailer. The system consists of an image gathering and processing subsystem, responsible for image gathering, image processing and control vector generation, as well as a tractor unit control subsystem, responsible for setting the speed and wheel turn in accordance with the received control vector. To automate the process of approach to a semi-trailer, among other methods, image recognition based on using ArUco markers was used [10]. This solution was implemented in order to speed up the process of image processing on the Raspberry Pi microcomputer. The research carried out proves that the solutions used are correct. Based on the experience gained, the plan is to develop a system which would support drivers during the approach of a real tractor unit to a semi-trailer. In the future, another round of semi-trailer type recognition tests is also planned, on image gathered directly from a camera, without using ArUco markers.

References

1. Agachai Sumalee, Hung Wai Ho: Smarter and more connected: Future intelligent transportation system, IATSS Research, Volume 42, Issue 2, Pages 67-71, 2018, ISSN 0386-1112
2. Andersen, J., Sutcliffe, S.: Intelligent Transport Systems (ITS) - An Overview, IFAC Proceedings Volumes, Volume 33, Issue 18, p. 99-106, 2000, ISSN 1474-6670
3. Benalla, M., Achchab, B., Hrimech, H.: Improving Driver Assistance in Intelligent Transportation Systems: An Agent-Based Evidential Reasoning Approach, Hindawi Journal of Advanced Transportation, 2020
4. Pena-Gonzalez, Raul & Nuño-Maganda, Marco Aurelio: Computer vision based real-time vehicle tracking and classification system, Midwest Symposium on Circuits and Systems, 2014
5. Bułatowa, I., Kamiński, R.: Rozpoznawanie tablic rejestracyjnych pojazdów na obrazach statycznych, Autobusy. Technika, Eksploatacja, Systemy Transportowe, 2016, ISSN 1509-5878
6. Zauraiz Alamgeer, Sathish Kumar Selvaperumal, Sophea Prum, Thang Ka Fei: Review of Car Make & Model Recognition Systems, Journal of Applied Technology and Innovation, Volume. 2, Issue 2, 2018

7. Kupiec, J., Kupiec, A., Kuśmierczak, M.: Problematyka eksploatacyjna sprzęgu siodłowego, Logistyka, nr 6, p. 6391-6398, 2014
8. Project website *wUZtruck*: http://www.wuztruck.imei.uz.zgora.pl/ downloaded: 07.05.2021
9. Jędrzejczak O.: System sterujący modelem ciężarówki podczas podjeżdżania do naczepy, Praca dyplomowa, Uniwersytet Zielonogórski, 2021
10. Czech, P., Mazurkiewicz, M., Mróz, P., Pławiak-Mowna, A.: Recognition of Semi-trailers on the basis of the image 2019, Proceedings of the 7th Prague Embedded Systems Workshop, Roztoky u Prahy, Czechy, Prague, p. 11-19, 2019

Satisfiability Modulo Simulation: What Can It Do?

Jiří Khun, Jan Schmidt
Faculty of Information Technology, CTU in Prague
Thákurova 9, Prague 6, Czech Republic

{jiri.khun,jan.schmidt}@fit.cvut.cz

Abstract. We analyze the situation where reasoning about a system is required, yet parts of the system can be only simulated. Several tools integrated a simulator with a SMT solver in various ways, creating Satisfiability Modulo Simulation. The problem, what questions such tools can answer in principle, remained open. We aim at such an analysis, based on the lazy solver architecture. We studied modifications that in general are required to accommodate a simulator. We identified the so-called domain propagation in the domain-specific part of the solver as the crucial process defining the properties of the entire solver. It is a Constraint Propagation problem. Based on the outcome of its solution, the solver can or cannot prove unsatisfiability. Also, various strategies to employ simulation experiments are possible. In the end, we show that the way to Satisfiability Modulo Black Box Evaluator is open.

Keywords. Satisfiability Modulo Theories, verification, simulation

1 Introduction

The presented paper is an attempt to analyze the situation, where we want to reason about a system using Satifiability Modulo Theories (SMT), but for some reasons outlined below a part of the system can be only simulated. We analyze, how simulation intracts with common techniques. For that, we need to briefly summarize their definitions. For introductory text on SMT and examples, please see [1] or other texts.

1.1 Verfication

The most common verification method is *simulation*, where a *simulator* interprets the verified design and provides its response to chosen input stimuli, in a series of *simulation experiments* or sessions. Because exhaustive simulation of all possible stimuli of practical systems is not possible, the quality of verification strongly depends on the design of simulation experiments.

Formal verification overcomes this problems. Using a formal description of the design, it *proves* the behavior of the design. The ultimate goal is to prove that the design conforms its formally expressed specification. This is called *equivalence checking*. Easier – and more frequently used – is to prove a certain property, again formally specified. The term *model checking* is correct but slightly misleading. The proof states that the design models – conforms to – a theory, that is, the verified property. Model-based engineering thinks of the design as a model of the future device, hence the confusion.

The verified properties usually fall into two categories. Either, that something good always happens, for example, that every operation finishes in a specified time. Or, that something bad will never happen, such as producing a dangerous output. Those examples are typical, the categories are often called *liveness* and *safety* properties, although in principle they are broader. In software verification, those properties can be linked to the state of the program.

1.2 Formulas

Many formalisms can be used to reason about a design. This paper deals with techniques where the design, the specification, and the checked properties are described using suitable logic, that is, as *formulas* in a logic, evaluating to *true* or *false*. As a good trade-off between expressivity and decidability, first order logic theories are used.

1.3 Questions and Problems

Let us have a formula, which captures the relationship between the input and output of the design. Another formula can describe admissible inputs to the verified system, and yet another can characterize the response of the system, e.g., whether it is dangerous. Using combinations of such formulas, various questions can be answered. For simplicity, we presume that the formulas themselves contain no quantifiers.

Standard verification asks: is there an admissible input that causes dangerous output? Formally, is there a valuation of formula variables such that the formula evaluates to *true*? If it can be *proven*, that it cannot happen, then the design is correct in this aspect. This way, we get from question to a *problem*. The problem discussed is the classical *Unsatisfiability Problem* (UNSAT) [2], and the proof that the formula is *unsatisfiable* is crucial. Note the existential quantifier is introduced by the decision problem of SAT. If the model checker proves satisfiability, it provides a counterexample.

When verifying a good, desired property instead, the question is different: for all admissible inputs, does the formula evaluate to *true*? Here, the ability to prove that the formula is *satisfiable* is important, the problem is not satisfiability but tautology (a problem complementary to UNSAT). Here, the quantifier is universal. If the tautology cannot be proven, an inferior strategy can be used: to try and find input values that *violate* the property. If this is successful, then certainly the design is incorrect; if not, nothing can be said for sure. Such tools are called *falsifiers*.

Yet another question is ”Is there a partial valuation such that, for all the remaining valuations, it holds...”. This question may ask, for example, whether there is a set o system parameters such that the system works for all inputs. It leads to the Quantified Boolean Formula problem [3].

1.4 Problems and solvers

Different questions lead to different problems, and for those problems, we need *solvers*. The important thing is that in some applications, *complete solvers* able to state SAT/UNSAT are needed, while in other situations, also *incomplete solvers* providing SAT/INCONCLUSIVE are also useful.

1.5 Logic theories

So far, we have not discussed the kind of logic of the formulas. They have to evaluate to *true* or *false*, so the logic must contain those constants, and possibly Boolean operators. For binary systems, this is all we need. However, even most digital systems process numeric values, and to capture continuous behavior, as in cyberphysical systems, we need real numbers. It is possible to construct logic theories with variables and operators in these and many other domains. As the resulting formulas are a combination of Boolean logic and another logic theory or theories, we get the problem of *Satisfiability Modulo Theories* (SMT) [4][5]. The formulas in these problems (SMT formulas) contain parts from the Boolean logic and from another first order theory, and any solver for such formulas is called an *SMT solver*.

1.6 SMT and Simulation

Mosterman et al. [6] studied model-based engineering in the above mentioned domains. They report that at first, the systems are described formally, e.g., by differential equations, and their parameters are estimated. To perform simulations, however, the estimated parameters must be changed to overcome inaccuracies and limitations of the simulator. Then the formal model is no longer the primary representation of the design, rather, the simulation model is primary. Worse yet, the simulation model depends on the simulator. So, formal reasoning about such a model cannot exclude the simulator, which the becomes a 'sacred cow' of all processing.

Let us underline that this is a practical approach. Therefore, we prefer practical considerations, such as computational requirements, to theory, such as decidability, in further text.

There are many tools that create a logic theory to suit a particular simulator or simulator class and which incorporate the simulator into satisfiability provers. Luteberget et al. [7] design a system that uses a combination of a SAT-based planner and an event-driven simulator to verify that a railway design fulfills given criteria. Kolárik and Raschan [8] formalize the simulation of ordinary differential equations in the domain of floating-point numbers and present a simple solver for that theory.

To use existing powerful and sophisticated simulators for formal reasoning is tempting. The crucial problem is, what kind of tasks can such systems solve, what kinds of questions can they answer. As this problem seems not to have been solved in its generality, this paper is an attempt to make first steps toward the solution. In Section 2, we outline what we understand as simulator and what we consider its principal features. In Section 3, we bring a more technical view of an SMT formula, and recall existing architectures of the solvers. This enables us to suggest a general architecture of a solver with connected simulator, and to analyze the roles of the simulator, giving insight into the class that can be solved by such a system.

2 Simulators and Simulation Experiments

We understand a simulator as a system which predicts the *response* of another system – the *simulated object* – to external conditions, the *stimuli*. A simulator performs a run, or *simulator experiment* under a degree of external control, and reports the results. In industrial use, the simulated object is augmented by other subsystems, often summarily called a *test bench*. These subsystems can generate stimuli and evaluate the response.

2.1 Simulator Variables

The stimuli can be input to the simulator directly or can be generated by an algorithm internally according to parameters from input. We consider these possibilities just technical variants, and call all of them *stimuli specifications*. Similarly, an *object specification* can be composed of descriptions in a language, parameter sets and so on. In principle, it does not matter whether response evaluation is done in the simulator or externally. We presume that any values of interest (e.g., signal values in time) can be obtained from the simulation experiment.

When dynamic systems are simulated, there is an important *simulation variable* – time. In other cases, the simulation variable can be frequency (the frequency response of a circuit) or space (e.g., the response of a 3D object to static load).

2.2 Control and stopping conditions

The stopping conditions are commonly included in the stimuli specifications or in the response evaluation. A more evolved communication schema can be found in electric mixed signal simulators. The digital part is event-driven, while the analog part simulates differential equations. Any of the simulator parts can be the main, controlling part. In the case where the digital part controls, it evaluates signals to the analog part, and runs the analog part whenever those signals change. Simultaneously, it tells the analog part when to stop; usually it is the case when the signals from the analog part of the circuit to the digital part change. It is interesting to notice that both the simulator keep the state of the respective parts of the circuit for further simulation. Examples of such communications can be deduced from the languages for mixed signal simulations, namely Verilog-AMS [9], VHDL-AMS [10], and SystemC-AMS [11]. In these cases, the conditions are on the simulation variable (time), or on response (threshold value).

2.3 Simulation Experiments

All simulators define the level of completeness required to run an experiment. In most cases, it means that the object and stimuli specifications must be complete. The cases where incomplete specifications can be accepted are rare; the most common one are the 'undefined' and 'don't care' values in logic simulation.

2.4 Simulator authenticity

For reasons explained in [6], using the same simulator as in the rest of the design process is necessary. As we show in further text, it would be beneficial to have the simulation more tightly integrated. The question is, how much deviation can a given application tolerate. For example, using a differential equations simulator, do we need, in comparison with the original simulator,

- the same computational model (differential equations),
- the same method and the same order (e.g., 4th order Runge-Kutta),
- the same step control,
- the same arithmetic (e.g., 64-bit IEEE754)?

The answers form a 'sacredness' or *fidelity* scale, and the actual position on that scale affects the solver in multiple ways:

- How much of the formula can be evaluated outside of the simulator?
- How much reasoning can be performed (e.g., deducing the initial conditions of a dynamic system from its final state)?
- Can we parallelize or accelerate the simulation?

3 SMT Solvers

There are two base methods to solve an SMT problem (see Nieuwenhuis et al. [1]). Here we will focus on the *lazy* method, as this is the only one that can accommodate an external simulator.

Let $B = \{false, true\}$ be the set of Boolean values in the usual first-order theory, with operators $B \to B$ and $B \times B \to B$. Let D be a non-empty set of values, and let us have a first-order theory for it, with operators $D \times \ldots \times D \to D$. As we are interested in satisfiability, the combined formula over B and D must evaluate to B. An example of a parse tree is shown in Figure 1. To construct such a formula, we need at least one operator $D \to B$, more commonly $D \times \ldots \times D \to B$ (domain *predicates*). There can also be operators $B \to D$, however, we omit them for clarity.

The operators $D \times \ldots \times D \to B$ form a natural 'demarcation line' between the part that works with Boolean values (we will call it the *SAT solver*), and the rest working with values from D, the domain solver or the *T-solver*.

The line also forms the communication path between the two parts. In the simplest case, the SAT solver prescribes the values of the operators, the T-solver finds values for variables in D and reports

SAT. If this is not possible, it returns UNSAT together with an *explanation*, that is, a set of clauses for the SAT solver to learn (the Clause Learning technique [12]). More complex communication schemes are possible; both the solvers can work incrementally, from smaller subsets to the complete formula. To reuse learned information, both parts should retain their state, which can be demanding for the T-solver. For complete explanations, again, see [1].

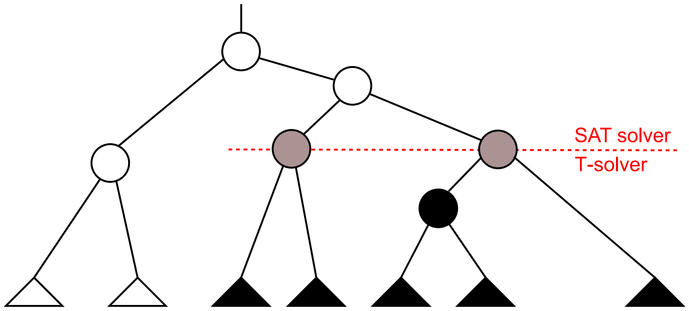


Figure 1: The tree of an SMT formula. Variables are triangles, operators are circles. Objects in B are in white, objects in D are in black, operators $D \times D \to B$ are in gray.

4 A T-solver With a Simulator

The values prescribed for the domain predicates form an assertion over D. The T-solver has to prove or disprove this assertion *experimentally*, using simulation experiments. The simulator has the role of *formula evaluator*. Such a top-level architecture is shown in Figure 2.

4.1 T-solver architecture

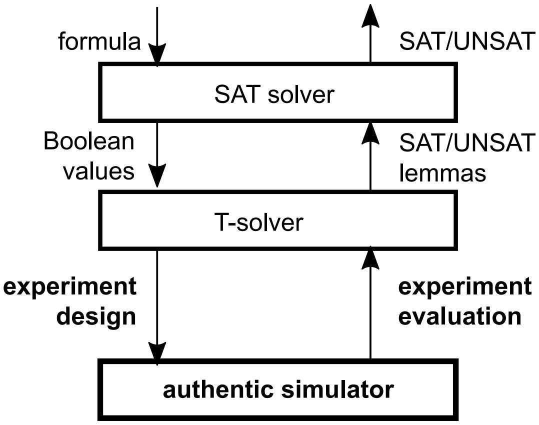


Figure 2: The top-level T-solver architecture using a simulator.

A standard feature of T-solvers is *domain propagation*, that is, inferring variable values from the prescribed valuations of domain predicates. In recent solvers, the propagation is *exhaustive* [1], which means that as many values are inferred as possible.

In a simulation-based T-solver, the inference is done mainly by simulation experiments. To run a simulation experiment, the input variables of the experiment must have concrete values. This immediately implies that at least some inference must be done outside of the simulator. For example, if x_0 is an initial value for a differential equation, and $x_0 = 5.5$ (or $eq(x_0, 5.5)$, to stress it is a predicate) should evaluate to *true*, then it can be done and has to be done before simulation. Therefore, the first step is domain propagation. Then, the rest of the inference is organized into simulation experiments. Finally, the experiments are run and evaluated, as in Figure 3.

The inference in domain propagation seems trivial; however, we have the authenticity requirements mentioned above. The predicates should evaluate as if done in the simulator. A well-known example is the equality of floating point numbers, especially in the case of lower precision.

The simulation variable or variables have a special position. In many cases, it is relatively easy to discover, such as time in dynamic systems described by differential equations. Stopping conditions for the simulator are frequently – but not always – connected to this variable. Again, it is up to the simulation experiments strategy to generate them.

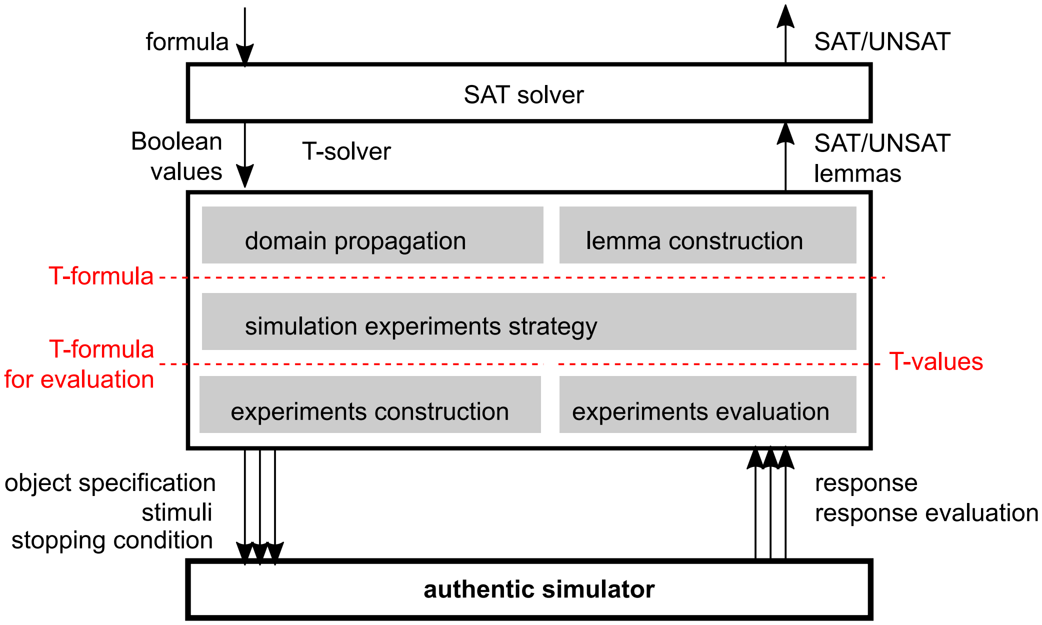


Figure 3: Detailed T-solver architecture using possibly multiple simulation experiments.

4.2 Experiment Design as the Constraint Propagation Problem

The design of a simulation experiment means, that simulated part of the system, the stimuli to it and the method of response evaluation are chosen. To keep the simulation experiment as simple as possible (or possible at all, see later), other parts of the formula need to be evaluated. They constrain the input values to the experiment, that is, the initial phase is constraint propagation. Constraint Propagation [13] is a thoroughly studied problem, with many methods available. The main idea is to give each variable its own domain, and then to reduce these domains to make them consistent with all constraints.

In our situation, the constraints follow from prescribed values. For example, if $eq(x_0, y_0)$ shall evaluate to *true*, then $x_0 = y_0$ is a constraint. Notice that the constraint is *not* directional; given a value of x_0, we infer y_0 *and vice versa*. Constraints that follow from parts of the formula evaluated by simulation, however, can be different. For example, we are not aware of any simulator that can work backwards in time, deriving stimuli from response. Such a constraint is therefore directional.

Intuitively, the more propagation happens, the easier task remains. The predicate $eq(x_0, y_0)$ constraints more when it evaluates to *true* than in the other case. For $noteq(x_0, y_0)$, the opposite holds. Therefore, the predicates can hint the SAT solver, as suggested by Kolárik and Ratschan [14].

The outcome of constraint propagation is characterized by cardinalities of the resulting domains. If the variables describing inputs to simulation are determined, then the set of simulation experiments (or one experiment, often) is also determined. This can be achieved already by limiting the properties of the original formulas.

It may happen that the input variables are not uniquely determined, but the cardinalities are so small that the brute force approach simulating all combinations of values is feasible. Here, we have to work with practical feasibility again – to compute the simulation may take any time from microseconds to weeks. Both the experiment strategies – with a single simulation and with multiple but feasibly few simulations – described so far lead to a complete solver, which can prove UNSAT.

If such a strategy is not feasible, we can still search for satisfying valuation heuristically. The outcome then could be SAT or INCONCLUSIVE, still useful e.g., in falsifiers.

In general, more than one simulation experiment is needed. This can be used to *parallelize* the solver. The constraints in the Constraint Propagation problem define data dependencies between simulations. Independent simulations will then be able to run in parallel.

4.3 Experiment Design for Infeasible Input Domains

When the domains of the input variables for simulation remain too large for exhaustive search, incomplete heuristics can be used. It would be beneficial to know that e.g., the search space has only one minimum, or that the response changes monotonically in a certain stimuli range. However, few if any simulators provide such guarantees. We have to use search methods that do not need any presumption on search space properties. As an less known example, let us mention simulated annealing adapted to continuous domains [15].

4.4 Simulators and CEGAR Procedures

Counterexample-Guided Abstraction Refinement (CEGAR) [16] is a stepwise refinement method used with many contexts. In conjunction with SMT, the formula is abbreviated first. Consequently, it has more satisfying values than the original formula. If the abbreviated formula is unsatisfiable, then so is the original. If a counterexample is found, then it may be just a result of the abbreviation, and the values can be used to change the abbreviation.

For a simulator-based *T*-solver it means, that the result of one experiment implies the result of another. In standard simulator use, such a guarantee is not required and therefore it is not given. One exception is the simulation time, as simulators are deterministic. In [7], formula abbreviation means shortening the time window. If no consistent train schedule can be found for an initial, shorter time window, then the same holds for a longer window starting at the same time. In this case, the simulator behaves identically in both cases, and to presume implication is reasonable.

5 Satisfiability Modulo Black Box Evaluator

Our notion of simulator (Section 2) is quite broad. It produces a response to stimuli, effectively evaluation a (sub)formula. The simulation variable and stopping conditions do not play a crucial role. Therefore, anything that can provide output values from input values, can play this role, as a *black box* (Figure 4a). If the simulator is ‘sacred’, then its input also may be considered ‘sacred’, and the design is seen as a black box. This is the highest-fidelity point on the scale mentioned in Section 2, and we already know that we have to sacrifice some reasoning capabilities for it. However, we still need logic to formulate requirements, properties, and questions. Therefore, the black box input and output still has to be characterized by a first-order theory (Figure 4b). Moreover, an impenetrable description prevents experiments with anything but the entire design. A compromise would be to presume that the original design is, e.g., modular and that subsets of the design can be chosen for experiments (Figure 4c).

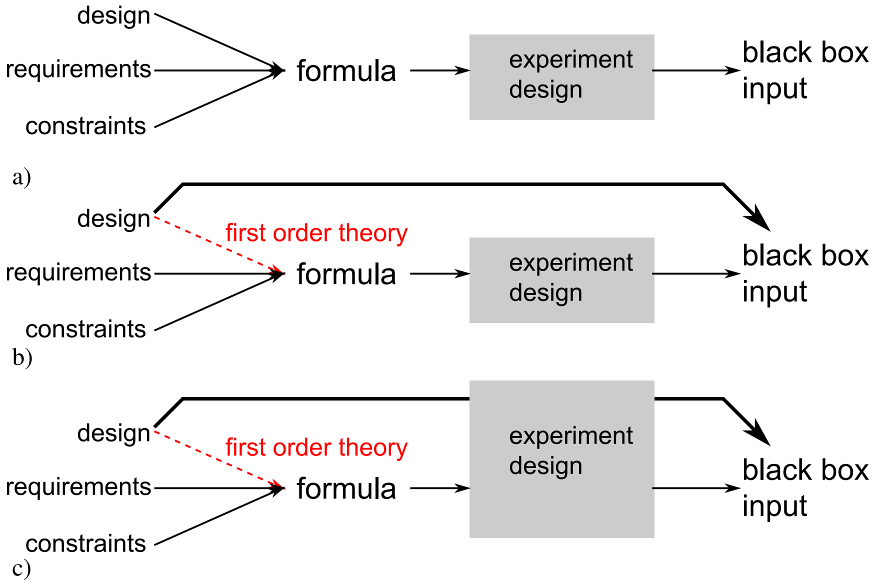


Figure 4: Data flow in a T-solver with an integrated black box: a) standard flow, b) with experiments on the entire design, c) in the case the design description can be parsed.

6 Conclusion

The domain propagation is a crucial part in a T-solver based on simulation. The outcome of constraint propagation, namely the cardinality of the resulting consistent domains, dictates the design of simulation experiments and hence the completeness of the solver. If the simulator is not strictly required to be authentic, ways to acceleration and parallelization are open. To be able to evaluate predicates outside of the solver, it is important to observe how such evaluation would be done in the simulator. Almost any guarantee about the behavior of the simulator can contribute to the T-solver. Also, the simulator can be replaced by any procedure or system that can evaluate the given type of formulas. Both complete and incomplete solvers can be designed in that manner.

Acknowledgment

We are grateful to Stefan Ratschan and Tomáš Kolárik at CTU in Prague for their patient explanations and valuable discussions. This work in fact originated from our effort to understand the full range and meaning of their research. Last but not least, we thank the reviewers for their thorough reviews.

References

[1] R. Nieuwenhuis, A. Oliveras, and C. Tinelli, “Solving SAT and SAT modulo theories: From an abstract Davis–Putnam–Logemann–Loveland procedure to DPLL(T),” *J. ACM*, vol. 53, no. 6, pp. 937–977, Nov. 2006, ISSN: 0004-5411. DOI: 10.1145/1217856.1217859. [Online]. Available: https://doi.org/10.1145/1217856.1217859.

[2] M. R. Garey and D. S. Johnson, “Computers and intractability; a guide to the theory of NP completeness,” in New York, USA: W. H. Freeman & Co., 1990, p. 338.

[3] M. Janota, W. Klieber, J. Marques-Silva, and E. Clarke, “Solving QBF with counterexample-guided refinement,” in *Proceedings of the SAT'12*, 2012. [Online]. Available: https://www.cs.cmu.edu/~wklieber/papers/qbf-cegar-sat-2012.pdf.

[4] H. Ganzinger, G. Hagen, R. Nieuwenhuis, A. Oliveras, and C. Tinelli, “DPLL(T): Fast decision procedures,” in *Computer Aided Verification*, R. Alur and D. A. Peled, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 175–188, ISBN: 978-3-540-27813-9.

[5] A. Biere, M. Heule, H. van Maaren, and T. Walsch, “Handbook of satisfiability,” in IOS Press, 2008, ch. 9.

[6] P. J. Mosterman, J. Zander, G. Hamon, and B. Denckla, “A computational model of time for stiff hybrid systems applied to control synthesis,” *Control Engineering Practice*, vol. 20, no. 1, pp. 2–13, 2012, Special Section: IFAC Conference on Analysis and Design of Hybrid Systems (ADHS'09) in Zaragoza, Spain, 16th-18th September, 2009, ISSN: 0967-0661. DOI: https://doi.org/10.1016/j.conengprac.2011.04.013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967066111000797.

[7] B. Luteberget, K. Claessen, and C. Johansen, “Design-time railway capacity verification using SAT modulo Discrete Event Simulation,” in *2018 Formal Methods in Computer Aided Design (FMCAD)*, 2018, pp. 1–9. DOI: 10.23919/FMCAD.2018.8603003.

[8] T. Kolárik and S. Ratschan, “SAT modulo differential equation simulations,” in *Tests and Proofs*, W. Ahrendt and H. Wehrheim, Eds. Springer International Publishing, Jun. 2020, pp. 80–99, ISBN: 978-3-030-50994-1. DOI: 10.1007/978-3-030-50995-8_5.

[9] Accelera Systems Initiative, *Verilog-AMS language reference manual 2.4*, online, 2014. [Online]. Available: https://www.accellera.org/images/downloads/standards/v-ams/VAMS-LRM-2-4.pdf.

[10] E. Christen and K. Bakalar, “VHDL-AMS-a hardware description language for analog and mixed-signal applications,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 10, pp. 1263–1272, 1999. DOI: 10.1109/82.799677.

[11] IEEE, *IEEE 1666.1-2016 - IEEE standard for standard SystemC(R) analog/mixed-signal extensions language reference manual*, 2016.

[12] A. Biere, M. Heule, H. van Maaren, and T. Walsch, “Handbook of satisfiability,” in IOS Press, 2008, ch. 4.

[13] R. Dechter, *Constraint processing*. Morgan Kaufmann, 2003, ISBN: 1-55860-890-7.

[14] T. Kolárik, “Satisfiability modulo simulated ODE,” unpublished presentation at a departmnental meeting at CTU in Prague, 2021.

[15] A. Corana, M. Marchesi, C. Martini, and S. Ridella, "Minimizing multimodal functions of continuous variables with the "simulated annealing" algorithm—corrigenda for this article is available here," *ACM Trans. Math. Softw.*, vol. 13, no. 3, pp. 262–280, Sep. 1987, ISSN: 0098-3500. DOI: 10.1145/29380.29864. [Online]. Available: https://doi.org/10.1145/29380.29864.

[16] E. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith, "Counterexample-guided abstraction refinement for symbolic model checking," *J. ACM*, vol. 50, no. 5, pp. 752–794, Sep. 2003, ISSN: 0004-5411. DOI: 10.1145/876638.876643. [Online]. Available: https://doi.org/10.1145/876638.876643.

Sponsors

Czech Technical University in Prague

The conference has been sponsored by the CTU grant SVK 56/21/F8.

EATON

Eaton’s mission is to improve the quality of life and the environment through the use of power management technologies and services. We provide sustainable solutions that help our customers effectively manage electrical, hydraulic and mechanical power – more safely, more efficiently and more reliably. Eaton’s 2019 revenues were $21.4 billion, and we sell products to customers in more than 175 countries. We have approximately 95,000 employees.

SYSGO

SYSGO is the leading European provider of real-time operating systems for critical embedded applications. Our products have been designed to meet the highest requirements when it comes to Safety and Security. Our customers are leading players in the Avionics & Defense, Space, Railway, Automotive and Industrial Automation and Medical industries, who use our PikeOS product as a platform for critical systems that need to be certified against industry-specific Safety and Security standards.

STMicroelectronics

STMicroelectronics is a world leader in providing the semiconductor solutions that make a positive contribution to people's lives, today and into the future. ST is a global semiconductor company with net revenues of US$ 8.35 billion in 2017. Offering one of the industry's broadest product portfolios, ST serves customers across the spectrum of electronics applications with innovative semiconductor solutions for Smart Driving and the Internet of Things. By getting more from technology to get more from life, ST stands for life.augmented.

CESNET

CESNET is an association of universities of the Czech Republic and the Czech Academy of Sciences. It operates and develops the national e-infrastructure for science, research and education which encompasses a computer network, computational grids, data storage and collaborative environment. It offers a rich set of services to connected organizations.

ASICentrum

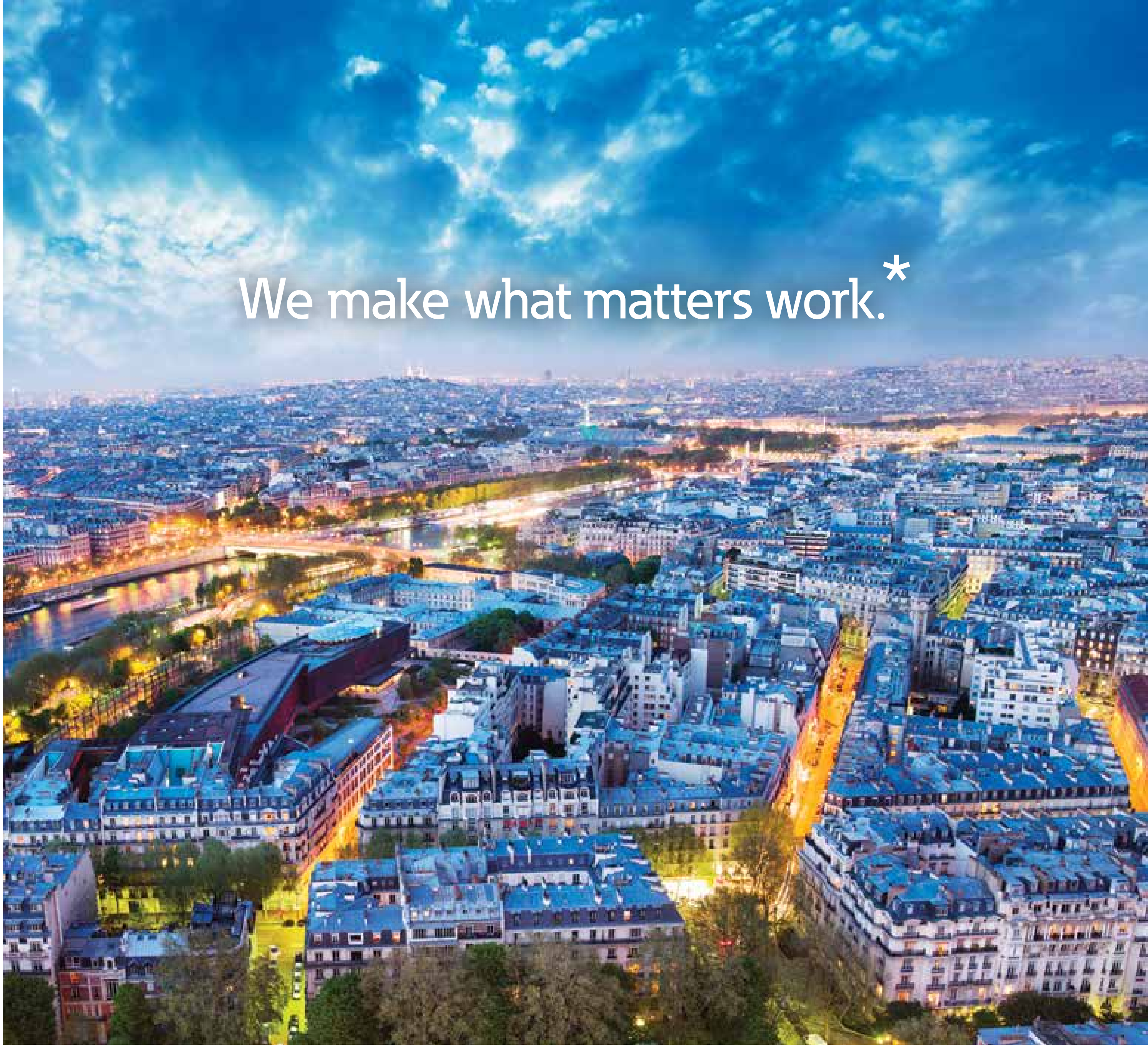
ASICentrum, established in 1992 in Prague is a design center of EM Microelectronic and a competence center of ETA, belonging to the Swatch Group. EM Microelectronic is one of the most innovative IC providers. It developed and manufactured the smallest and the lowest power consuming Bluetooth chip on the market, the top performing optical sensors for optical office as well as gaming mice and it was the first to release the award-winning world-first dual-frequency NFC + RAIN RFID emlecho.

Partners

IEEE Student Branch at Czech Technical University in Prague

IEEE Young Professionals

Computer (C) Society Chapter of the Czechoslovakia Section of IEEE


We make what matters work.*